AD-A280 641

Concepts of Fuzzy Model Assessment

P. R. Kersten
Combat Control Systems Department

S. C. Nardone
University of Masachusetts Dartmouth

DTIC
ELECTE
JUN 2 7 1994
S
G
D

94-19459

Naval Undersea Warfare Center Division
Newport, Rhode Island

DTIC QUALITY INSPECTED 5

94 6 24 086

PREFACE

Reviewed and Approved: 1 March 1994

P. A. La Brecque
Head, Combat Control Systems Department

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 1 March 1994	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE Concepts of Fuzzy Model Assessment	5. FUNDING NUMBERS

6. AUTHOR(S)

P. R. Kersten
S. C. Nardone

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Undersea Warfare Center Division Newport, RI 02841-1708	8. PERFORMING ORGANIZATION REPORT NUMBER TR 10,342

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report discusses the concepts and mathematical models needed to apply fuzzy system theory to contact management model assessment. The detection and classification of propagation models are essential components of contact tracking. The classification of propagation models is a pattern recognition problem, which is addressed in this report by using fuzzy system theory. The sensor data are modeled with fuzzy numbers, the decision rules are constructed using fuzzy rules, and the decision quality is evaluated by interval-valued certainties. This system is one key component to building a fuzzy expert system tracker.

14. SUBJECT TERMS			15. NUMBER OF PAGES 52
Fuzzy Sets Pattern Recognition Fuzzy Logic Fuzzy Experts Fuzzy Expert Systems			16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR

TABLE OF CONTENTS

i

LIST OF ILLUSTRATIONS

LIST OF ABBREVIATIONS AND ACRONYMS

bf	Base frequency change
bpa	Basic probability assignment
BIOFAM	Binary input/output fuzzy associative memory
CRI	Compositional Rule of Inference
CLOS	Common Lisp Object System
CMMA	Contact management model assessment
CDF	Cumulative density function
FLOPS	Fuzzy Logic Official Production System
FNN	Fuzzy neural network
FPR	Fuzzy pattern recognition
FS	Fuzzy system
h0	Null hypothesis or no change in model
h2	Contact maneuver
inf	Infimum
LPC	Linear piecewise continuous
LISP	List processing language
MF	Membership function
MPG	*Modus ponens* generating function
NN	Neural network
nl	Unknown anomaly
PDF	Probability density function
pp	Propagation path
ROC	Rule of combination
s-norm	Triangular conorm
SPR	Statistical pattern recognition
sup	Supremum
SNR	Signal-to-noise ratio
t-norm	Triangular norm

CONCEPTS OF FUZZY MODEL ASSESSMENT

1. INTRODUCTION

When tracking a contact via passive sonar, one finds the solution structure of the track is in large part determined by the stationarity assumptions made on the stochastic processes associated with the tracking solution. Stationarity is assumed for the noise processes, the kinematics of the two vessels, and the acoustic channel linking the source and the observer, which in our model corresponds to the contact and own ship, respectively. Essentially, one is forced to assume the processes are piecewise stationary, i.e., stationary for periods of time with the change points between stationary periods occurring quickly with respect to the expected length of stationarity. During periods of stationarity, a filter can track the contact. Between the periods of stationarity, the filter may lose track and should in any event be reinitialized after the switching period between stationary periods. Thus, one has a renewal process with periods of stationarity during which tracking is possible. The points of renewal in this process are the ends of the transition periods of stationarity.

To successfully model this process, one must not only design the tracking filters for the stationary periods but also detect the change points and determine the new propagation model that will hold during the next period of stationarity. The essence of model assessment is the classification of the possible propagation models. Model assessment is a pattern recognition problem, which can be approached in many ways. Here a fuzzy system is used to determine the possible models and the confidence associated with each model. From this information, a decision is made as to which possible models should be maintained in constructing the tracking solution during the next period of stationarity. This report describes the application of fuzzy system modeling to propagation model assessment, with the emphasis on theoretical issues.

1.1 DATA ABSTRACTION OF THE TRACK

The piecewise stationarity of the observation process is reflected in the tracking solution. The overall tracking solution is a string of tracks built up over many periods of stationarity. A track established on a single period of stationarity is called a segment. For a single contact, the piecewise construction of the solution leads to the hierarchical representation of the tracking solution as a tree. For a set of contacts, the solution is a forest of trees. Segments are classified according to mechanism that caused the segment formation. The mechanisms include the following:

1. Change of propagation path (pp) (e.g., a change from a direct path to a bounce path between source and observer or vice versa). In fact, any change in the propagation channel fits into this category including multi-bounce models.

2. Changes in the own ship heading, speed, or depth.

3. Changes in the base frequency of the source (bf), which should produce minor changes in propagation channel characterization provided the change in frequency is a small percentage of the base frequency.

4. Changes in the contact heading, speed, or depth (h2), which are usually observable in the Doppler and bearing measurements.

5. Changes in the system of unknown type (nl). Unknown anomalies are a way of specifying the class of changes not modeled so far.

Segments not caused by contact maneuvers are used to build up contact segments, or those parts of the track where the contact has not changed its kinematics. A sequence of contact segments is then used to forms a contact track, which is represented as a tree in this hierarchy. The set of all contact tracks is a collection of trees or a forest of trees, which is the contact situation assessment.

This solution construction is illustrated in figure 1-1, where on the right side of the figure a single physical contact track is drawn and on the left side is the corresponding data abstraction that represents the ideal solution for this track. In the physical track, the small white circles represent data points in the x-y plane.
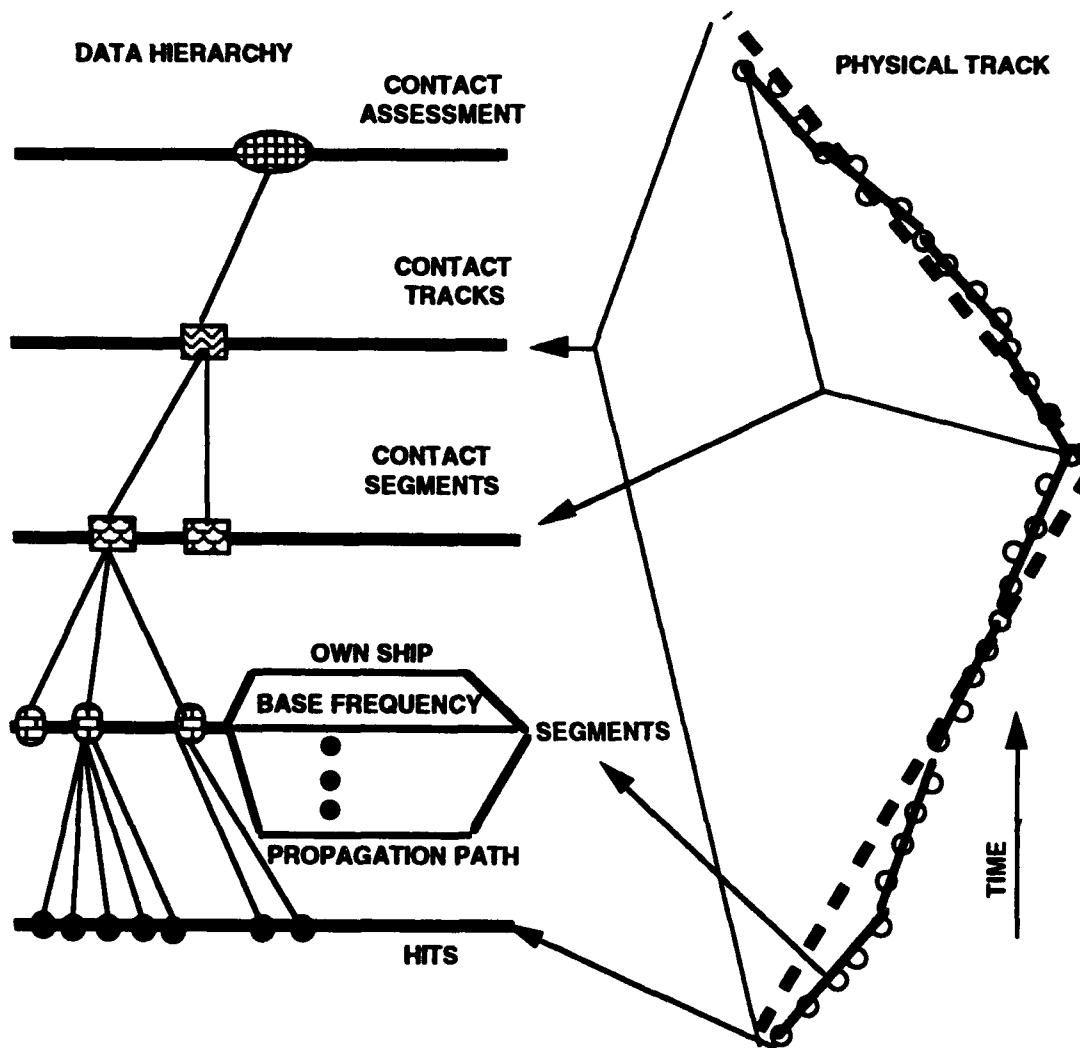
Figure 1-1. Solution Hierarchy Induced by the Piecewise Stationarity of the System Processes

Tracks formed during stationary periods are called segments and drawn as short solid lines fitted to the data points. The linear fit is the result of the tracking process or state estimation. This fit is sometimes called feature extraction, data reduction, or data abstraction. As long as the contact does not maneuver, the segments can be viewed as data points in a "super segment" called a

contact segment. Contact segments are drawn as a bold dashed lines in the physical track and there are two contact segments in figure 1-1 where the track evolves in time from the bottom of the figure to the top. The contact segments are concatenated to form a single physical track in this figure. Since the physical track is built up by successive data abstraction, a natural data hierarchy emerges as shown on the left-hand side of figure 1-1. The most basic data elements or data points form the first level of the hierarchical structure. Proceeding up the data hierarchy, groups of points generated during track formation produce the segments, groups of segments form contact segments, groups of contact segments form contact tracks, and at the top of the hierarchy, the set of all contact tracks form the contact assessment. Each level of the hierarchy has a corresponding physical meaning shown by the arrows.

For the tracker to perform successfully, the system must be able to detect the change points in the stochastic processes that are affecting the system, and then classify them so that one can reinitialize the tracking filters properly. This later task, if successful, allows a savings of computational resources because the alternative is to initialize a new filter for every possible scenario and see which ones converge. Previous efforts in this field have used Dempster-Shafer evidential reasoning and compatibility maps that map observed effects to possible causes and, at the same time, produce a consistent basic probabilistic assignment (bpa) to the possible causes. Reference 1 describes one such effort. Although this subtask is critical, it is by no means the only issue in designing a Fuzzy Expert System. This report considers modeling issues relevant to the Fuzzy Expert System Tracker. Many of these issues are also relevant to the Contact Management Model Assessment (CMMA) simply because the tracker includes the CMMA functionality within the tracking mechanism. A more complete description on the CMMA system is contained in reference 2. This report emphasizes theoretical issues. The fuzzy CMMA system's description and performance is a topic for future work, once a suitable and correct data set is obtained.

1.2 CLASSIFICATION OF THE MODEL

Classifying the cause of change points is a pattern recognition problem, which can be formulated using many different methods. Statistical pattern recognition (SPR) is one and fuzzy pattern recognition (FPR) is another. Although there are many different techniques and approaches to pattern recognition, they all require (1) a correct model of the problem, (2) extraction of a set of features that describe the differences in the classes, and (3) construction of a decision rule to map the feature space to the decision space. Fukunaga (reference 3) describes the pattern recognition process using the flow diagram of figure 1-2. In this figure, the first three blocks represent the initial data collection, processing, and exploratory data analysis. The data structure block consists of the search for structure in the data using clustering and modeling techniques along with data reduction or feature extraction. Feature extraction is an art requiring iterative refinement measured by the error estimates achieved. Finally, the features are used as input to the classifier designed to fit the data for the anticipated use of the system. The final product needs to be tested to validate the design procedure. Note the number of loops and the emphasis on nonparametric methods to determine the performance of the classifier, before it is built. The flow diagram assumes a valid data set, which is not available at this point. So the blocks that address normalization, error estimation, and actual clustering and modeling of the features will have to wait until valid simulated data are available or real data is found. So in terms of Fukunaga's flow chart, this report can only address the data structure analysis block and the classifier design block. The approach is to replace the classical decision rules and the feature extraction by their fuzzy system counterparts, which are the fuzzy rule base and the fuzzy term sets, respectively. This issue will be discussed sections 2 and 3.

```
┌─────────────────────┐
│   DATA GATHERING    │◄────────────┐
└─────────────────────┘             │
           │                        │
           ▼                        │
┌─────────────────────┐             │
│   NORMALIZATION      │            │
│   REGISTRATION       │            │
└─────────────────────┘             │
           │                        │                ┌─────────────────────┐
           ▼                        │                │  ERROR ESTIMATION   │◄────────┐
┌─────────────────────┐             │                │  (NONPARAMETRIC)    │         │
│  ERROR ESTIMATION   │─────────────┘                └─────────────────────┘         │
│  (NONPARAMETRIC)    │                                         │                    │
└─────────────────────┘                                         ▼                    │
           │                               ┌──────────────────────────────────┐      │
           ▼                               │         CLUSTERING               │      │
┌─────────────────────┐                    │      STATISTICAL TESTS           │      │
│  DATA STRUCTURE     │◄────────►           │         MODELING                 │      │
│    ANALYSIS         │                     │                                  │      │
└─────────────────────┘                     │     FEATURE EXTRACTION           │──────┘
           │                               └──────────────────────────────────┘
           ▼
┌─────────────────────┐                    ┌──────────────────────────────────┐
│    CLASSIFIER       │◄────────►           │      DESIGN / REDESIGN          │
│     DESIGN          │                     │      LINEAR CLASSIFIER           │
└─────────────────────┘                     │      QUADRATIC CLASSIFIER        │
           │                                │      PIECEWISE CLASSIFIER        │
           ▼                                │    NONPARAMETRIC CLASSIFIER      │
┌─────────────────────┐                    └──────────────────────────────────┘
│  ERROR ESTIMATION   │                                ▲
│   (PARAMETRIC)      │─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─►┘
└─────────────────────┘
           │
           ▼
┌─────────────────────┐
│    FIELD TESTS      │
└─────────────────────┘
           │
           ▼
```

Figure 1-2. Fukunaga's Flow Chart of the Process of Classifier Design

The goal of pattern recognition is to design a decision rule that partitions the feature space into equivalence classes, one equivalence class per cause of change point. The extracted features are usually represented as vectors in a Euclidean space. Thus, the decision rule is a map from the observation space to the causes or to the description of the equivalence classes. So the decision fits the traditional representation of a decision rule as a map from the observation space to the set of integers $\{1,...,c\}$, where each integer represents a class ($D: X \rightarrow \{1,...,c\}$). The statistical decision rule is shown in figure 1-3a as a map that places a unit value at the "proper" class element and zero values in all the other elements. So for this example, $\delta(x) = 2$. In this figure, the one-dimensional observation space is partitioned into five disjoint regions, and the labeled areas are the characteristic functions associated with each of the regions. Classical sets may be defined in terms of a characteristic function that takes on the value 1 if the point is in the set and 0 if the point is not in the set, i.e.,

$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & otherwise \end{cases}.$$

For example, the second equivalence class has a characteristic function denoted by

$$X_2(x) = I_{(b_1,b_2]}(x), \text{ where } I_{(b_1,b_2]}(x) = \begin{cases} 1, & x \in (b_1,b_2] \\ 0, & otherwise \end{cases}.$$

For this example, the statistical decision rule is given by

$$\delta(x) = \sum_{i=1}^{5} iX_i(x),$$

where only one of the $X_i(x)$ can be non-zero. In the vector $u(x) = [X_1(x),...,X_5(x)]$, each coordinate is associated with a class and so each decision is represented by a unit vector. So the vector of classes for figure 1-3a is $u(x) = [0,1,0,0,0,]$. The fact that only one element of the function can be 1 also follows from the definition of partition. A point may be in one and only one partition and one and only one equivalence class, and thus in only one class. This is known as a "hard" decision where the word hard refers to the all or nothing quality of the decision. In figure 1-3a, the sample x can belong only to one class, class 2. Conceptually, the decision rule $\delta(x)$ and the vector $u(x)$ merely represent the decision in different formats. In the latter format, $\delta(x)$ can be thought of as the index of the non-zero element in the vector $u(x)$, and each element of $u(x)$ can be interpreted to mean the degree that the data point belongs to the class associated with that index. In fact, the vector $u(x)$ represents the characteristic function of the class chosen by the decision rule $\delta(x)$, so in effect, the decision can be thought of as a map from the observation space to the space of characteristic functions representing the classes. Thus one could recast the decision rule as a mapping $D: X \rightarrow \{0,1\}^C$, where $\{0,1\}$ is the two-element set containing 1 and 0.
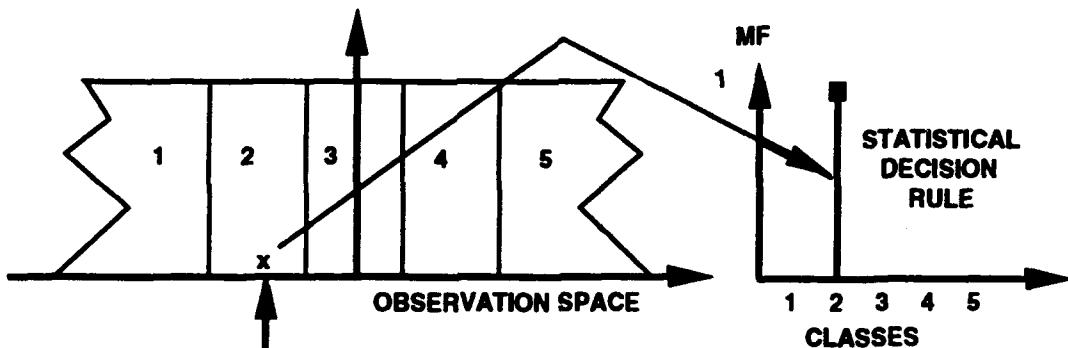
Figure 1-3a. Decision Rule in Statistical Pattern Recognition

By replacing the partition of the data space by a soft partition, one can construct soft decision rules. In the previous paragraph, a decision rule was interpreted as a mapping to the set of characteristic functions. Characteristic functions reflect the philosophy of classical set theory: a point either belongs to a set (takes value 1) or it does not (takes value 0) - there is no in-between. Similarly, fuzzy decision rules can be described as mappings to the set of membership functions (MFs) of fuzzy sets. MFs generalize characteristic functions and define fuzzy sets. When evaluated at a point, a MF determines the degree that the point belongs to the set. For a fuzzy set A, the MF is denoted by $\mu_A(x)$, and takes on values in [0,1]. In SPR, a decision rule

partitions the observation space as shown in figure 1-3a, where each partition is a subset of the observation space and that subset is defined by a characteristic function, say $\chi_2(x)$ as discussed above. In a fuzzy partition, a point in the observation space can be in any one of the classes with varying degrees of membership. The memberships are constrained by the requirement that they sum to one for each point in the observation space, i.e.,

$$\forall x \in X, \quad \sum_{i=1}^{c} \mu_i(x) = 1 .$$

The decision is no longer a mapping into the set of integers $\{1,...,c\}$ but instead, into the vector space $D: X \to [0,1]^C$, just as the hard decision rule may be thought of as a mapping to the vector space $D: X \to \{0,1\}^C$, where $u(x) = [\chi_1(x),...,\chi_c(x)]$ represents the map. By softening the hard decision rule so that one has a fuzzy partition of the observation space, the decision rule is interpreted to be a vector-valued map to a fuzzy unit vector (fit vector), which is one representation of the MF of a fuzzy set. Here, $u(x) = [\mu_1(x),...,\mu_c(x)]$ is both the decision and the fit vector. Each element of the fit vector is the membership of the data point in the class denoted by the coordinate index. One class exists for each dimension of the vector. In the notation of fuzzy sets, the fit vector can be represented by $\mu(x) = \mu_1(x)/1 + \cdots + \mu_c(x)/c$, where each pair $\mu_i(x)/i$ is the membership of the point in class i: one class for each dimension of the vector. Because the observation space partition is fuzzy, the vector of classes is no longer a unit vector, but only a vector whose elements sum to 1.

Using the same observation space of figure 1-3a, consider the fuzzy partition shown in figure 1-3b. Now the classes are not disjoint and overlap exists between the classes. The partition of the observation space is fuzzy and is defined by the fuzzy sets, one for each class. As noted, the fuzzy sets are the generalizations of the characteristic functions defining the classical partition of figure 1-3a. In a fuzzy partition, a data point represented by x in this figure, may support a decision with varying degrees in class 2 or class 3 or both. Thus, each possible decision has associated with it a membership value, which represents the truth that the data supports that decision. And in fact, the decision rule can be generalized further, so that the sum of the elements in the vector of classes does not have to sum to 1, but only that each element of the vector is bounded above by 1. In summary, SPR produces a single unambiguous decision, whereas FPR produces an MF that evaluates for each class to the belief that the data belong to that class.
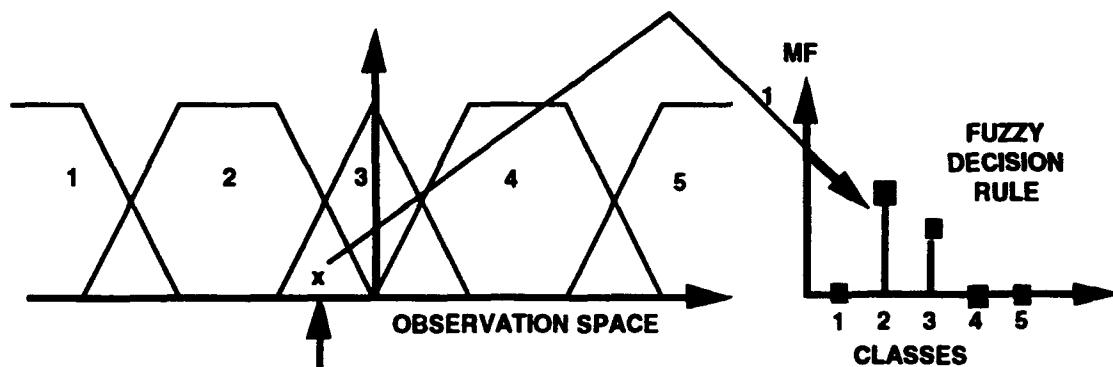

Figure 1-3b. Decision Rule in Fuzzy Pattern Recognition

1.3 EXTENSION OF THE CLASSIFICATION PROCESS

This report takes the above concepts one step further where the decision rule maps from the fe1ture space to interval-valued sets, one set for each class. So each element of the class vector is no longer a real number between 0 and 1, but instead a closed subset of the interval [0,1]. Figure 1-4a illustrates this decision rule for the same observation space of figure 1-3a. Classes 2 and 3 have some measure of support, but now that support is represented by an interval. In general, all the classes can have interval-valued support and the general form of the decision rule is $\mu(x) = \mathcal{X}_{A_1}(x)/1 + \cdots + \mathcal{X}_{A_c}(x)/c$, where A_i is a closed interval. Each interval represents the range of truth the data can support the decision for that class, where truth is measured on a scale from zero to one. The data are represented by fuzzy numbers illustrated by the shaded region in the observation space. This vector of intervals is described in the literature as a fuzzy set of type 2 and called an interval-valued fuzzy set (reference 4, p. 14). The interval-valued measure of truth is a generalization of FPR. Because the decision rule maps to a closed interval, which itself can be represented by a characteristic function or an MF, one can illustrate the decision rule as shown in figure 1-4b. Conceptually, no difference exists between figure 1-4a and 1-4b; the only change is the representation of the certainty intervals yielded by the decision rule. The decision system described in this report extracts features that are fuzzy sets, uses a collection of fuzzy rules to replace the decision rule of SPR, and uses interval-valued fuzzy sets to represent the output of the decision rule. Note that each interval in this figure may be the result of many rules. Each rule tries to assess the support a feature gives to some class. If an unambiguous decision is required, then further processing is needed. The important point is that the entire pattern recognition process is extended to produce not just a single membership value for each class, but a collection of MFs represented by interval-valued fuzzy sets.
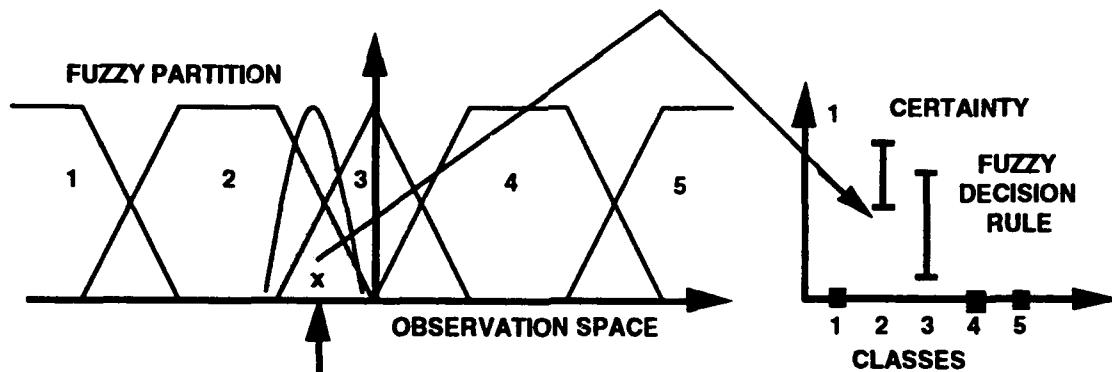


Figure 1-4a. Decision Rule with Interval-Valued Certainty
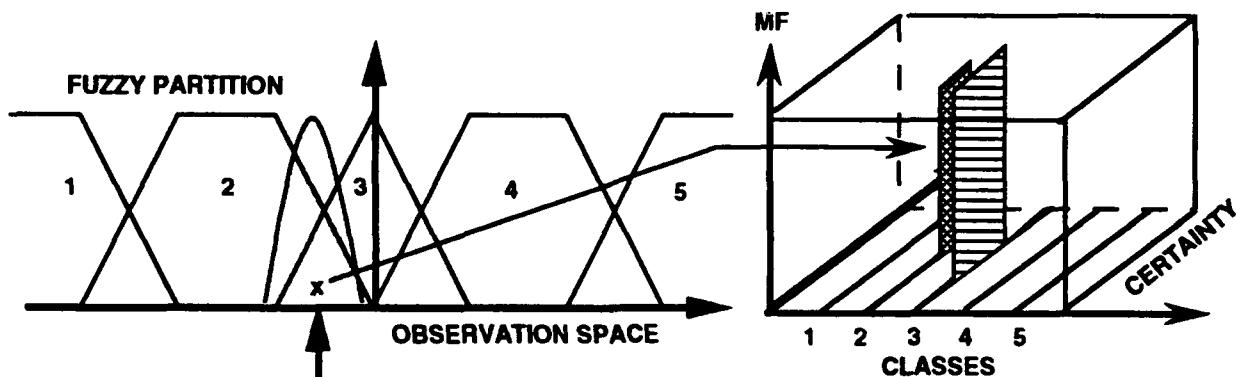
Figure 1-4b. Decision Rule with Interval-Valued Memberships

In the following sections, fuzzy logic is applied to generate a decision rule of the form $\mu(x) = \chi_{A_1}(x)/1+\cdots+\chi_{A_c}(x)/c$ and then applied to model assessment in contact tracking. In classical decision theory, the decision rule is derived by optimizing a functional such as the probability of error or another loss function. In this report, the decision rule is never explicitly stated because the certainty intervals are the result of a fuzzy rule base that models the governing expert decision system. In section 2, fuzzy systems are discussed in more detail to explain how the fuzzy features are extracted from the data and processed by fuzzy rules to yield a decision that is interval valued as shown in figure 1-4. In section 3, interval-valued fuzzy logic is described and used to solve the pattern recognition problem. This discussion includes the determination of the satisfaction of a fuzzy premise by a fuzzy feature, the representation of the strength of the rule, the propagation of evidence through the rules, and the aggregation of support for a given conclusion. In section 4, an exploratory version of a fuzzy model assessment system is discussed. Included in this discussion is the construction of the term sets, the rule syntax and implementation, the object-oriented aspects of the program, and examples of the outputs. In the last section, the implications of this work are given and possible extensions suggested.

1-8

2. FUZZY DECISION SYSTEMS

Fuzzy decision systems use fuzzy logic to reason about decisions. These systems represent signals as fuzzy numbers or linguistic variables and employ fuzzy rules to process these signals. In this section, a cursory description of linguistic variables is presented and the basic concepts of fuzzy systems are described. A simple control example is given to illustrate how fuzzy systems process the input signals to yield a control signal. This sample also gives the simplest illustration of a certainty measure associated with the conclusion. Fuzzy decision systems are compared with classical decision systems and a physical interpretation of the system flow is given.

2.1 EMBEDDING FUZZY SYSTEMS

To understand fuzzy decision systems, one must first understand the signals that flow through them, i.e., the linguistic variables. Formally, a linguistic variable is defined (reference 5, p. 132) as a quintuple $(x, T(x), U, G, M)$ where x is the name of the variable, $T(x)$ is the term set of the variable x, U is the domain of definition of the variable x, G is a rule that names the terms and M is the fuzzy set that is used to define each term. All this formality can be easily explained using an example. Consider the linguistic variable called COLOR, where the linguistic variable name is x = COLOR. Figure 2-1 illustrates this linguistic variable, where the term set $T(x) = \{$RED, YELLOW, ORANGE, GREEN, BLUE, INDIGO, VIOLET$\}$. The rule G that generates the names of the terms is just the list of basic colors but, in general, G can be a complicated grammar. The universe of discourse U or domain of definition is the frequency of the visible spectrum. Finally, the meaning or semantic definition of the terms is given in this example by a fuzzy set, drawn as a bell-shaped curve around the central frequency associated with the color.
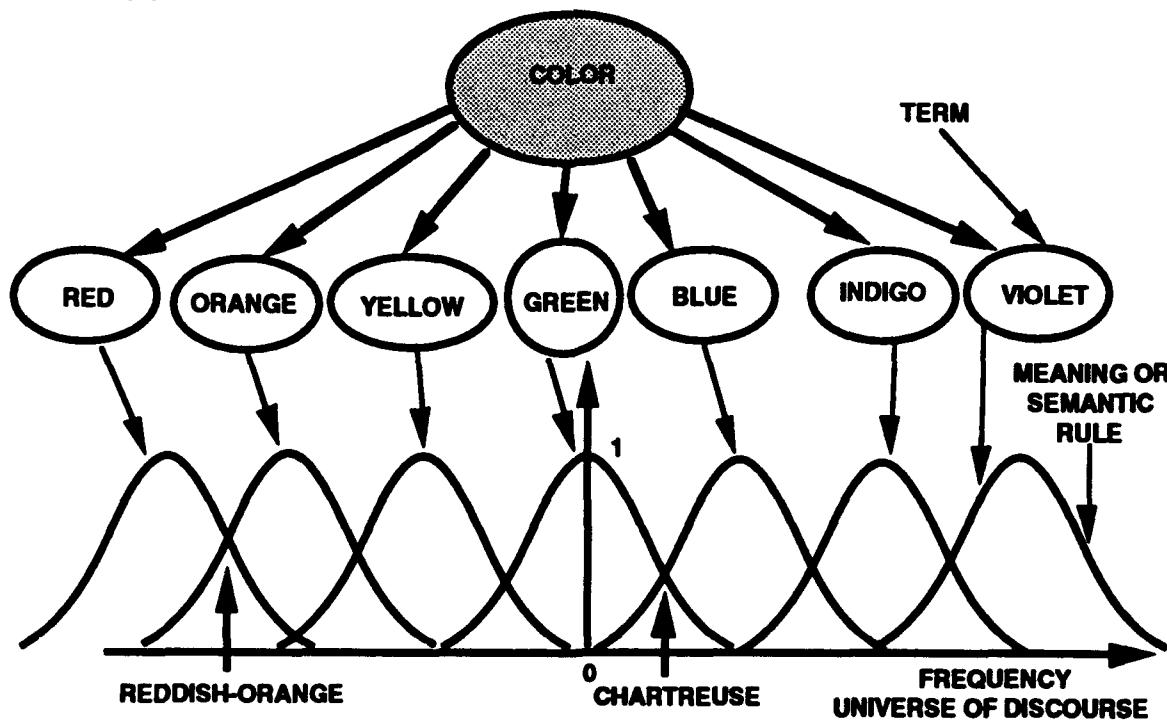

Figure 2-1. Linguistic Variable COLOR

The shape of the linguistic terms is important and are usually defined mathematically by the MFs such as $\mu_{RED}(x)$. Intuitively, this function $\mu_{RED}(x)$ is interpreted as the degree of membership of the frequency x in the set of RED colors, or how "red" is the frequency x. The range of the function $\mu_{RED}(x)$ is [0,1], where 0 means no membership in the set and 1 means total membership into the set of red colors. Partial membership is best illustrated by the example of a sunset - which is reddish-orange, having membership in both the RED and ORANGE terms. In figure 2-1, this frequency is labeled reddish-orange. Another example in this figure is the frequency labeled chartreuse, a frequency half-way between GREEN and BLUE. The terms of the linguistic variables can divide the space producing a fuzzy partition, i.e.,

$$\forall x \varepsilon X, \exists \, at \, least \, one \, y \, s.t. \, \mu_{T(y)}(x) > 0 \, \text{ and } \, \sum_T \mu_{T(y)}(x) = 1 \, ,$$

so that at every frequency at least one MF is non-zero and the sum of the MFs at that value of frequency is 1. Note the terms of the linguistic variable COLOR do not form a fuzzy partition because by inspection, at the frequency located between BLUE and GREEN, the sum of the membership values is less than one. Fuzzy partitions must then have special term sets.

Some of the concepts of a fuzzy system for model assessment are best described by using linguistic variables. Bezdek outlines these concepts best in his introductory article in the first issue of the *IEEE Transactions of Fuzzy Systems* (reference 6). Figure 2-2 illustrates the fuzzy system model used in this report and reference 6. In this figure, the data are first fuzzified, i.e., transformed or mapped into linguistic variables. The fuzzy system (FS) processes these linguistic variables by using rule-based methods and obtains one or more conclusions. The processing is carried out by the fuzzy inference engine using the rules stored, the fuzzy rule base where application-specific knowledge is contained in the data term sets. So the processing solves the problem. However, the rules yield answers that are in terms of fuzzy sets or in terms of conclusions with associated certainties. To apply the answers, one must be able to project back into the control or decision space. In a control problem, this space is a control value or vector. In a decision problem, this space is a finite set of decisions. The process of projection is called defuzzification. The four-step process is summarized by Bezdek as "fuzzify, solve, defuzzify, control" (reference 6, p. 3).
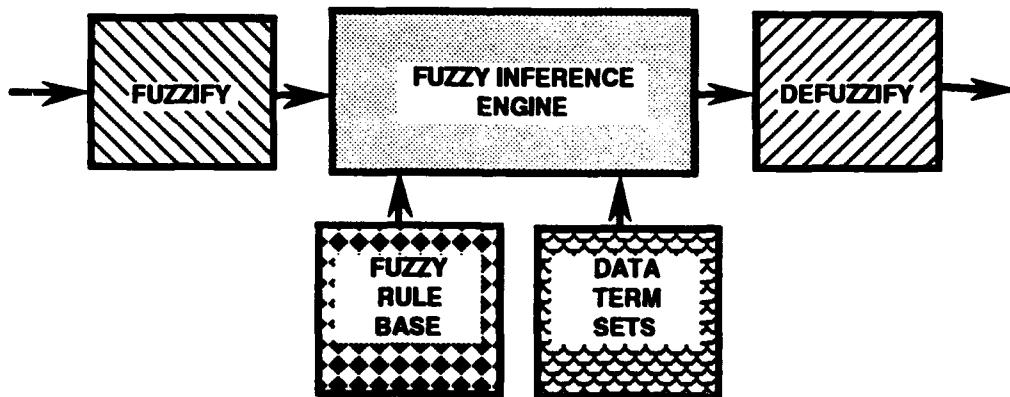
Figure 2-2. Fuzzy System Model

This solution process is similar to using Fourier transforms to solve a linear system. Here one transforms the input into the complex frequency domain, multiplies the input transform by the system transform, thereby solving for the output in the transformed space. However, to know what this output is, the inverse Fourier transform must be applied to return to the time domain:

transform, solve, and take the inverse transformation. Using transforms is similar to, but not the same as, the four-step process of fuzzy systems. Fuzzy systems are similar in that one transforms to another representation to make the problem easier, and then converts back from that representation. However, the Fourier transform is invertable in some sense because the inversion from the transformed space totally recovers the signal for a given class of signals.

The linguistic variables used in the fuzzy system are not just another representation of the input. They are in a sense, an embedding of the input into the space of linguistic variables. This process can represent numerical vectors or functions without loss of fidelity. Since linguistic variables can represent data exactly, one could think of these as embedding the input into this larger space. However, this is not usually the case. Once the data have been fuzzified, i.e., mapped into linguistic terms, usually some information has been lost so the input cannot be retrieved. The important system information has been extracted from the input, and this information is then used to solve the problem. The model trades representational complexity for solution tractability and robustness.

Embedding as a solution technique has long been used by mathematicians. An example is the evaluation of real integrals by complex analysis. Here one embeds the integrand in the complex number system, and the path of integration within the complex plane. The path of integration includes the real number line, so that one can apply the residue theorem in evaluating the integral on a closed path, thus obtaining the real integral as part of the entire solution. Here the portion of the integral along the real number line represents the solution and the process of projecting back onto the real number line becomes a triviality. The similarity is more evident when the process is described as complexify, solve, and decomplexify (reference 6).

For the system to solve the problem, linguistic variables are used to fire fuzzy rules. The rules are mappings from the transformed input space of linguistic variables to the solution space of linguistic variables. Multiple levels of rules can be thought of as a composition of mappings. Unfortunately, these mappings are multivalued. So for control problems, the rules can produce ambiguous results. In the case of decisions problems, the rules can produce a set of possible decisions. In binary logic, either a rule fires or it does not, depending on the premise being satisfied or not. But in fuzzy logic, all the fuzzy rules fire with varying levels of satisfaction of the premise and with varying levels of confidence in the implication, leading to conclusions with varying levels of belief. These levels of satisfaction, confidence, and belief are termed certainties, and the passage of confidence through the rules is called propagation of evidence or propagation of certainty through the rules.

With multiple answers at the output of the system, the control designer has to take the multitude of answers and disambiguate or interpolate between them to obtain a single value. This process in FSs is called defuzzification, which is analogous to the projection back into the solution space when solving by embedding. In decisions systems, multiple answers mean the designer must select a subset of the solutions and use these to further the decision process in some manner. Methods for handling ambiguity in the decision process is an active area of research.

The philosophy of a fuzzy system bears a resemblance to the linear systems concept of resolution of signals into sinusoids or impulse functions. In fuzzy systems, the representation or signal space is now linguistic, which resolves the signal into membership in the term sets of the linguistic variable. The solution technique uses fuzzy rules applied to the input term sets and yields an output fuzzy set or a set of decisions. In linear systems, the signal resolution is combined at the end of the analysis using superposition. In fuzzy systems, the output linguistic terms are aggregated in a non-linear fashion. The defuzzification of the output term often requires a functional to map the output to the solution space. The strength of fuzzy systems is that in some circumstances, fuzzy rules can model nonlinear systems easier than more

mathematically precise methods. The rules can be heuristic in nature and derived not just from the physical model of the system, but from the experts' knowledge of the behavior of the system learned over time. The learning need not be from just human experts, but can also be derived from learning methods such as neural networks (NNs).

Fuzzy systems are applicable to model assessment because humans can take the sensor readings and, using heuristics, determine what type of propagation path or model is now valid. Heuristics can be implemented as fuzzy rules and it is hoped that these rules will be as efficient and as accurate as human observers. The process is one of pattern recognition, i.e., given a set of tracking residuals and knowing the current track, determine the new propagation model. Although this can be modeled as a statistical pattern recognition problem, it becomes a hard decision problem, i.e., only one decision is accepted. However, this really does not make sense from an operational point of view. If the data do not support a clear and obvious model for the propagation, but instead a set of possible models, then all of these models should be pursued in parallel until further evidence resolves the ambiguity. Clearly, it is more efficient to resolve this ambiguity as soon as possible, and this is part of the reason for doing the work. However, if one discards the correct model by using a hard classifier, then the contact can be lost if the correct model is not used in the tracking process. The approach is to make a soft decision, clearly identifying the most promising model candidates with the hope that the true model is one of the model candidates.

2.2 FUZZY CONTROL EXAMPLE

The four-step solution sequence is "...fuzzify, solve, defuzzify, control" (reference 6). Fuzzification is based on the definition of the terms in the linguistic variables. For any data value or input x the nonzero terms t that have support, i.e., $\{t | t \varepsilon T(x), \mu_t(x) > 0\}$ are used to code or to fuzzify the data into the terms of the linguistic variable. Lotfi Zadeh, the father of fuzzy sets and systems calls this "fuzzy quantification." So in the example of the sunset, the terms supported by the reddish-orange hue are {RED, ORANGE}. For a sensor measuring some quantity such as speed, the terms may be {ZERO, POSITIVE SMALL}. It is important to remember that associated with each value x and with each applicable term t, a membership value exists describing the degree of membership in this term. This value is very useful as illustrated in the following control example.

A taxi driver who is approaching a red light, must brake to come to a safe stop. The control variable is the rate of braking and the two inputs are the speed of the car and the distance to the stop light. In this simple example there are only two variables and two fuzzy rules used to determine the braking rate. The two rules have the following form:

RULE 1:
IF the speed of the car is HIGH,
AND the distance to the stop light is NEAR,
THEN the braking should be HARD.

RULE 2:
IF the speed of the car is MEDIUM,
AND the distance to the stop light is MEDIUM,
THEN the braking should be MEDIUM.

Where the term set for speed is {SMALL, MEDIUM, HIGH}, the term set for the distance is {NEAR, MEDIUM, FAR}, and the term set for the braking is {LIGHT, MEDIUM, HARD, PANIC}. Both rules are illustrated in figure 2-3: the first rule is on top and the second rule on

the bottom. The terms are defined by triangular fuzzy sets that cover the three universes of discourse, i.e., speed, distance, and braking rate. The important point is that the fuzzy rules are approximations to an exact analytical description of the system. In this figure, the input speed x has membership in both terms {MEDIUM, HIGH} of the linguistic variable speed. The degree of membership is evaluated using these term sets. Similarly, the distance y to the red light has membership in {NEAR, MEDIUM}. The minimum value of the memberships of these two inputs is used to determine the "strength" of the conclusion, which is a term of the linguistic variable called braking rate.
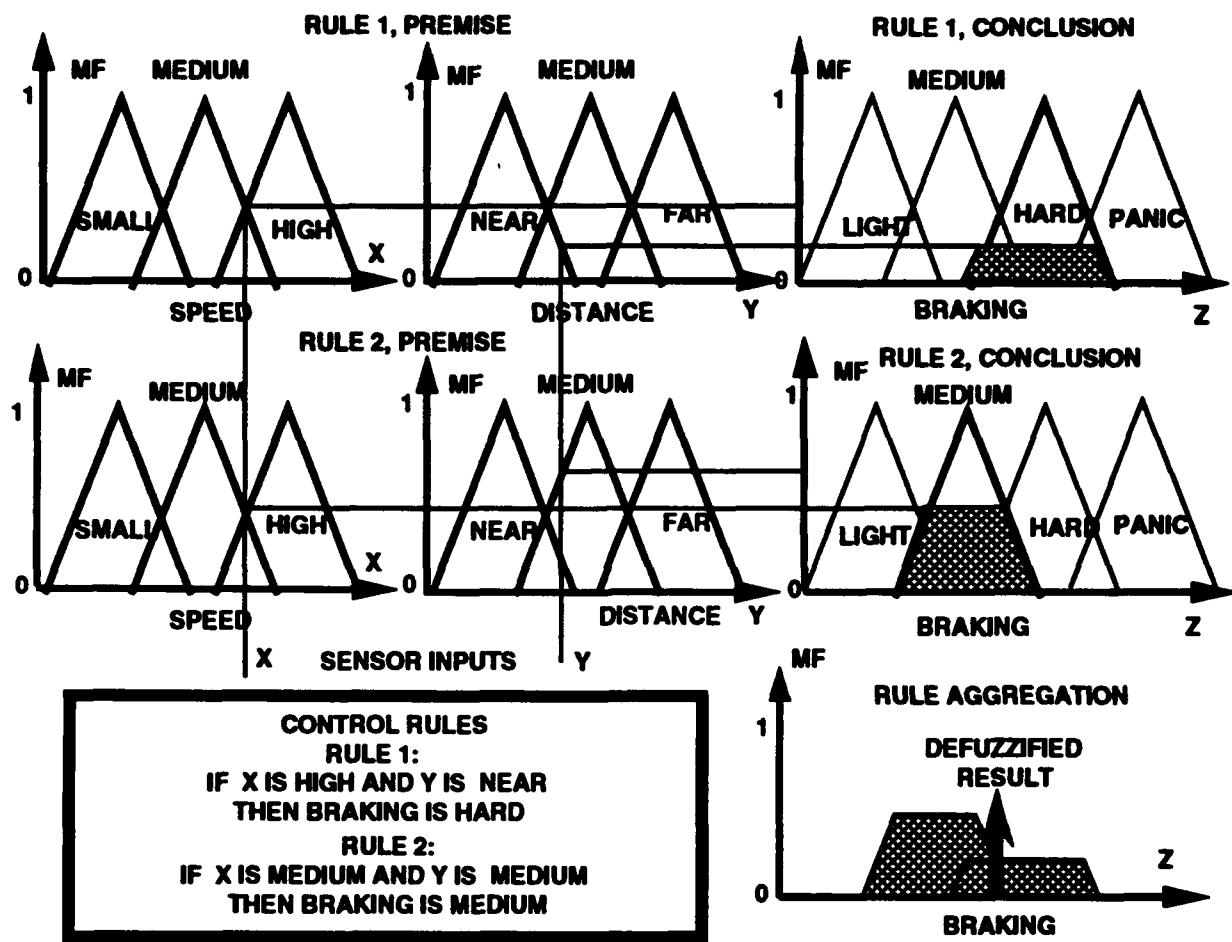
Figure 2-3. Typical Output Strength Calculation in Fuzzy Control Logic

The two braking rules yield different braking rate terms, which are aggregated to yield a single output braking rate. The aggregation procedure is called "defuzzification" and is a simple averaging of the areas in the output fuzzy sets. Again, the relative strength of the rule output is determined by the minimum of the memberships of the inputs in each of the input premises. In effect, the strength of the output for a rule is determined by the degree of satisfaction of the premise clauses. Premise satisfaction or certainty and its propagation through the rules are an inherent part of the solution technique. In fact, data are evidence only if they satisfy the premise as measured by the certainty. Propagation of the certainty through the ply to determine the certainty of the conclusion is the propagation of evidence through the fuzzy rule. The single value obtained in the solution space by defuzzification is the control part of Bezdek's "... fuzzify, solve, defuzzify, control."

Note in this example, the certainty of the premise or its validity or its degree of satisfaction is a single-valued real number. The strength of the premise is equivalent to the notion of the MF value or validity. However, certainty can be represented in other forms besides a single-valued real number between zero and one, such as interval valued and functional valued. In this report, the interval-valued version of the certainty is used, which is more general than the single-valued certainty illustrated in the control example. However, Bezdek's model still holds. The main change is in the certainty representation, which will be discussed in detail in section 3.

2.3 SOFT DECISIONS VERSUS HARD DECISIONS

In section 1, it was pointed out that a hard decision produces a single cause or conclusion and a soft decision produces either a list of possible conclusions or a list of conclusions with a relative ranking. Dempster-Shafer theory of evidential reasoning is an example of soft decision. The support for the soft decision is on the set of all subsets of the possible conclusions. So if the possible conclusions are {h0, pp, bf, h2, nl}, then the decisions of the form "bf-or-h2" are allowable decisions. SPR produces a hard decision since one is essentially doing an N-class test of hypothesis where the decision is one of N disjoint conclusions. SPR produces a single conclusion after the test is made. FPR can do the same thing if the decision process is based on Bayes classifier as described in reference 7. Here, fuzzy clustering is used to replace the standard statistical estimation of centroids and dispersion, and then these estimates are used in a maximum *a posteriori* decision rule. The answer is still "hard," only one decision value. Assume, as discussed in section 1, that one has a vector, $u(x) = [u_1, ..., u_C]$ then a "hard" decision is represented as a unit vector where only one component has a value of one, and all others have a value of zero. A data vector is represented by a boldfaced letter, e.g., x_k.

With unsupervised clustering, one learns about the structure of the data without the aid of labeled data or as is said in some cases without a teacher. As the data are clustered, each data point x_k has associated with it a fit vector $u(x_k) = [\mu_1(x_k), ..., \mu_c(x_k)]$, representing the memberships of x_k in each of the c classes. This vector can be thought of as a fuzzy unit vector or fit vector as called by Kosko (reference 8) or as a fuzzy set and represented by

$$\mu(x_k) = \mu_1(x_k)/1 + \cdots + \mu_c(x_k)/c ,$$

which is the standard fuzzy set notation. The only requirement placed on the components of this vector is that they sum to 1, i.e.,

$$\sum_{i=1}^{c} \mu_i(x_k) = 1.$$

As shown in figure 2-4, this means that for three dimensions, the vector is constrained to lie in the plane illustrated by the shaded part. For hard classifications, the constraint is even more severe; the membership vector or fit vector is required to be one of the unit vectors, also illustrated in figure 2-4. An application is the fuzzy k-nearest neighbor algorithm. One finds the

2-6

k-nearest neighbors, and then averages their membership vectors. The membership vector indicates to what degree the sample vector x_k belongs to each class. If one must make a hard decision, then the class with the maximum component will provide the hard decision. Ties can be broken randomly or arbitrarily.
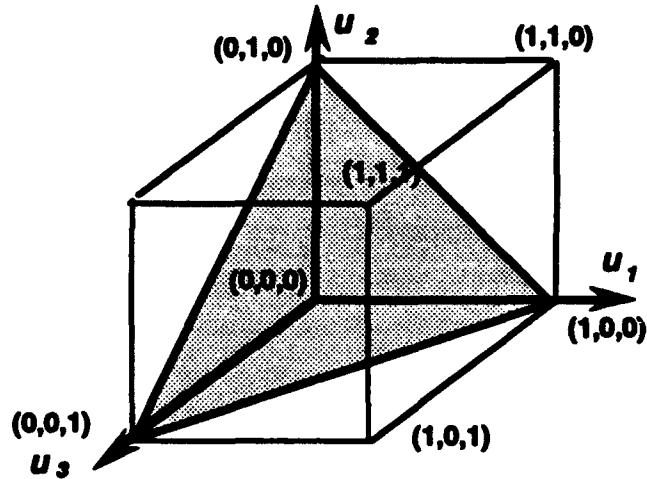

Figure 2-4. Hard Decisions vs Soft Decisions

The type of classification or pattern recognition occurring in model assessment is even more general than the fuzzy nearest neighbor. Associated with each class is a closed interval, which represents the lower and upper bounds on the membership in that class. So, the membership vector has now been replaced with what is called an interval-valued fuzzy set. Klir defines the interval-valued fuzzy sets as "... membership functions of the form: $\mu_A : X \rightarrow P([0,1])$ where $\mu_A(x)$ is a closed interval in [0,1] for each $x \epsilon X$ " (reference 4, p. 14). Here X, the universe of discourse of the fuzzy set, is one of the possible classes, and the interval associated with each class is the closed interval representing the lower and upper bound associated with the certainty of membership of each class. This situation is illustrated in figure 2-5, for a five-class problem. The fuzzy set is given by

$$\mu(x_k) = I_{A_1}(x_k)/1 + \cdots + I_{A_c}(x_k)/c,$$

where instead of single-valued membership functions of figure 1-3a, one has interval-valued membership functions as illustrated in figure 1-4. In this fuzzy set, the integers $\{1,\ldots,c\}$, $c = 5$ are the decisions and the universe of discourse. The support for each decision is a set represented by the indicator function $I_A(x)$, which is 1 if $x \in A$ and 0 elsewhere. For the interval-valued fuzzy sets, the set A has the special form $A = [a_1, a_2]$, where a_1 is the lower bound of the support and a_2 is the upper bound of the support. As in FPR, there is no clear-cut winner since each of the possible conclusions can have support. The process of aggregating this support will be discussed later.
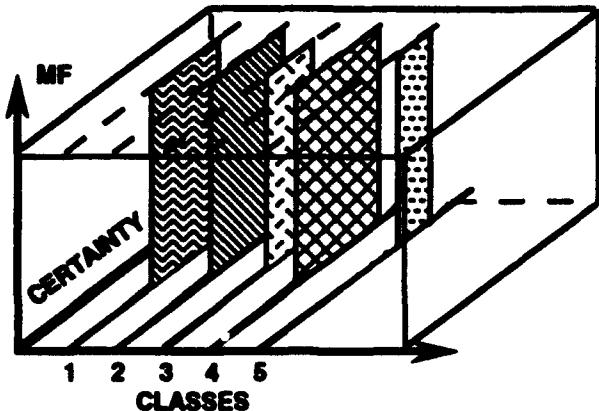
Figure 2-5. Interval-Valued Fuzzy Sets

2.4 CLUSTER INTERPRETATION OF RULES

Supervised clustering occurs when data samples that lead to a known decision are clustered. In effect, one is approximating the decision function by determining the pre-image of a known conclusion. The decision function then can be specified by $\delta(x_k) = i \Leftrightarrow x_k \varepsilon X_i$, where X_i is the i-th cluster. Essentially, these clusters are fuzzy rules mapping portions of the input space to integers in the output space. The integers are the classes that are referred to as models such as bf, h2, pp, h0 and nl, as well as the index into the vector of classes. Kosko points out that "The key idea is *cluster equals rule*" (reference 8, p. 330). Although Kosko was referring to a very specific case of a binary input/output fuzzy associative matrix (BIOFAM), the general idea holds here as well.

Figure 2-6 illustrates the formation of clusters or rules for model assessment. In this figure, the feature extraction from the data does the same thing that a Hough transform would, i.e., maps lines to points. However, here is a line represented by a two-dimensional point (intercept, slope). The slope is called the drift and the intercept is called the jump because of its historical reference to the bearing tracks. The feature space in figure 2-6 is a four-dimensional space, and the cluster are mapped to the decision called h2. The point is, that this cluster represents a region of the feature space that maps to an integer in the decision space. Intuitively, these mappings are the fuzzy rules.

In this report, features are extracted from the data to drive the decision system and the features are then modeled as fuzzy sets. Using the variance of each feature, a normal density is fitted to the feature. When renormalized, this density is interpreted as a fuzzy number that is treated as input data. It is this interpretation that allows the decision process to be modeled totally in terms of fuzzy constructs. In fact, the entire system is a fuzzy expert system where the data objects are fuzzy sets and the inference engine uses fuzzy logic. The fuzzy rules are viewed as composite mappings from the input space to the decision space. The steps are summarized as follows:

2-8
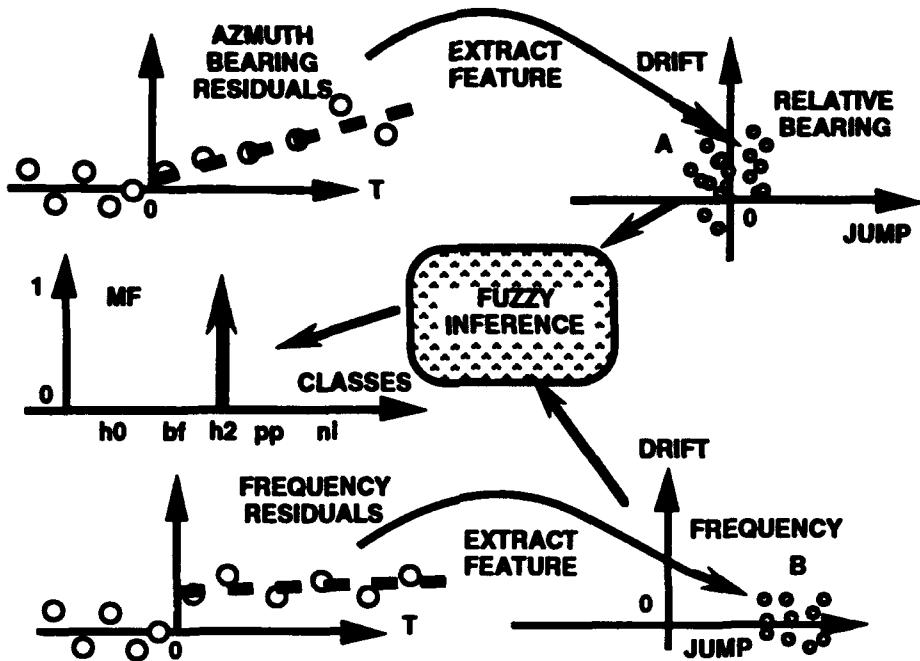
Figure 2-6. Formation of Clusters in Model Assessment

1. The data are fitted using linear regression techniques.

2. The features are extracted from the linear fit and their densities are re-normalized to construct fuzzy numbers. These numbers now form the input data to the fuzzy system whose general structure was illustrated in figure 2-2.

3. The clauses of the fuzzy rules and the fuzzy partition of the feature space are then used to construct closed certainty intervals associated with each clause. These intervals are, in turn, combined to form a premise certainty interval.

4. The premise certainty interval is mapped to the output decision space using the strength of the rule itself.

Figure 2-7 illustrates this mapping procedure. Note that the clusters illustrated in figure 2-7 are two-dimensional, but this is not how they are presently implemented. Instead, they are implemented as two one-dimensional clusters. This figure is for conceptual purposes only. As mentioned in section 2.2, the decisions are described in terms of interval-valued fuzzy sets. So conceptually, the fuzzy rule is a mapping from regions in the input space to the output space, but in this case, because of the form of the input data and the output decision, the mapping is harder to visualize. But conceptually, Kosko's claim remains, "...*cluster equals rule.*"
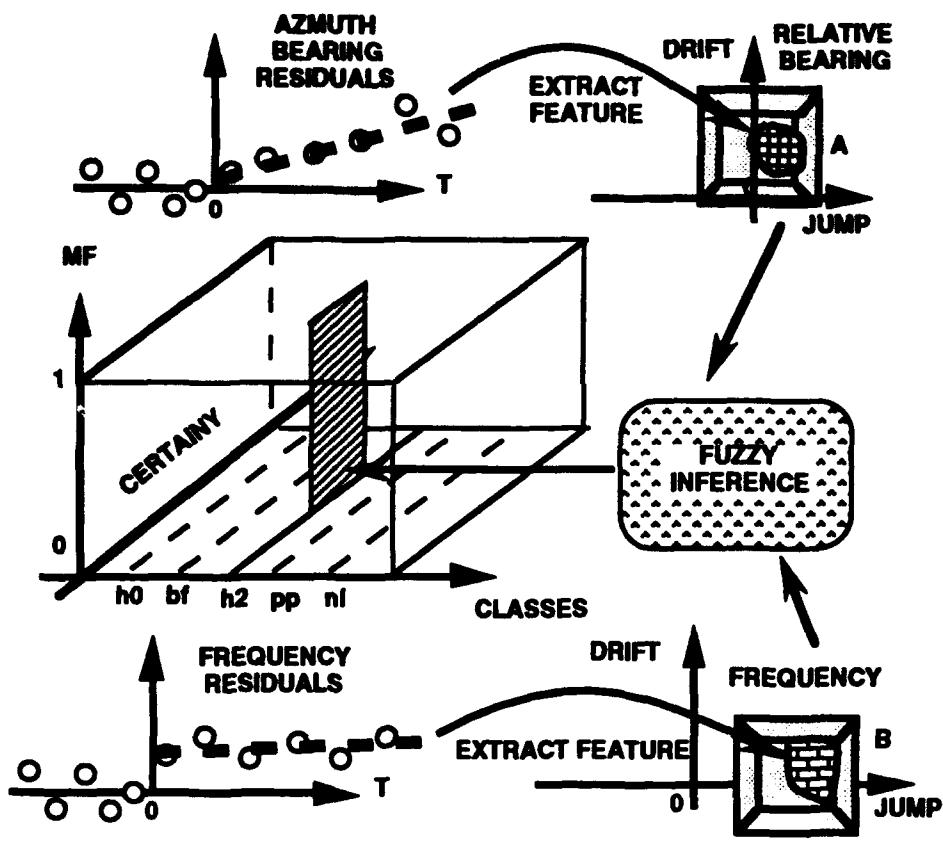
Figure 2-7. Fuzzy Rule Maps to Interval-Valued Fuzzy Set

3. FUZZY RULES IN CLASSIFICATION

3.1 FUZZY RULES

In section 2, fuzzy rules were described as mappings from a cluster or region of the feature space into the decision space. Propagation of certainty through fuzzy rules was also discussed and illustrated in a simple control example. The main components of Bezdek's description of a fuzzy system were illustrated in this example, namely, " ... fuzzify, solve, defuzzify and control." In the taxi driver example of figure 2-3, the certainty was represented as a real number in the interval [0,1]. In this section, the model is extended to include a more sophisticated method of certainty representation - an interval-valued representation. For the remainder of this report, only the interval-valued certainty representation is discussed. Intuitively, this interval can be thought of as lower and upper bounds on the satisfaction of an event, premise, conclusion, or fact.

3.1.1 Interval-Valued Certainties

First consider how interval-valued certainty arises from the interaction of the fuzzy data and the fuzzy rule. When the data is itself uncertain, the input is modeled as a fuzzy number, which is a special type of fuzzy set. A fuzzy set is represented by an MF, which is a generalization of the characteristic function used to express sets in real analysis. So it is a mapping from the domain of definition X to the interval [0,1] denoted $\mu_A(x):X \rightarrow [0,1]$. A fuzzy number is a convex normalized fuzzy set (reference 4, p. 17), normalized meaning the MF has a maximum of 1, or $\sup_{x \varepsilon X} \mu_A(x) = 1$. Intuitively, the convexity of a fuzzy set implies that it looks like a bell-shaped or single-mode curve. Formally, convexity of fuzzy sets is defined in terms of α-cuts of the MF, which are crisp sets representing the region of X with membership greater than or equal to α, i.e., the crisp set defined by $A_\alpha = \{x \mid x \varepsilon X, \mu_A(x) \geq \alpha\}$. A fuzzy set A is convex if and only if all of its α-cuts are convex. In figure 3-1, both the data and the property are convex sets, and this makes the derivation of the certainty interval easier.
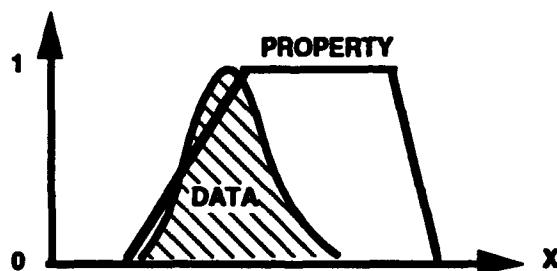
Figure 3-1. Fuzzy Data Fitted to a Fuzzy Premise

Certainty intervals are derived in CMMA from the fact that the input data are fuzzy numbers, and the properties defining the clauses are represented as fuzzy sets. Again, in figure 3-2 one sees the property as $\mu_B(x)$ and the data as $\mu_A(x)$, except here only one term set is shown. Conceptually, the upper bound of the certainty interval represents the amount of overlap of the fuzzy data A and the fuzzy property B. In possibility theory, the standard intersection of two fuzzy sets is given by the minimum function, i.e.,

$$\mu_{A\cap B}(x) = \min_{x\varepsilon X}[\mu_A(x),\mu_B(x)],$$

and the maximum of $\mu_{A\cap B}(x)$ represents the largest degree of overlap. This minimum of two fuzzy sets is the counterpart to the logical AND. The standard union of two fuzzy sets is given by

$$\mu_{A\cup B}(x) = \max_{x\varepsilon X}[\mu_A(x),\mu_B(x)].$$

The union is the counterpart of the logical OR and complementation is the counterpart of the logical NOT. Complementation is represented as $\mu_{\overline{A}}(x) = 1 - \mu_A(x)$. Both of these concepts are needed to construct the possibility and the necessity. The lower bound of the certainty interval is represented by the measure of subsethood of fuzzy set A in fuzzy set B, or the degree that A is a subset of B. Mathematically, the upper bound is defined as the possibility:

$$\Pi = \sup_{x\varepsilon X} \min[\mu_A(x),\mu_B(x)].$$

The lower bound is defined as the necessity:

$$N = \inf_{x\varepsilon X} \max[1 - \mu_A(x),\mu_B(x)].$$

The graphical construction for both of these measures is illustrated in figure 3-2.

$$POSSIBILITY = \sup_x min[\mu_A(x),\mu_B(x)] \quad NECESSITY = \inf_x max[1-\mu_A(x),\mu_B(x)]$$
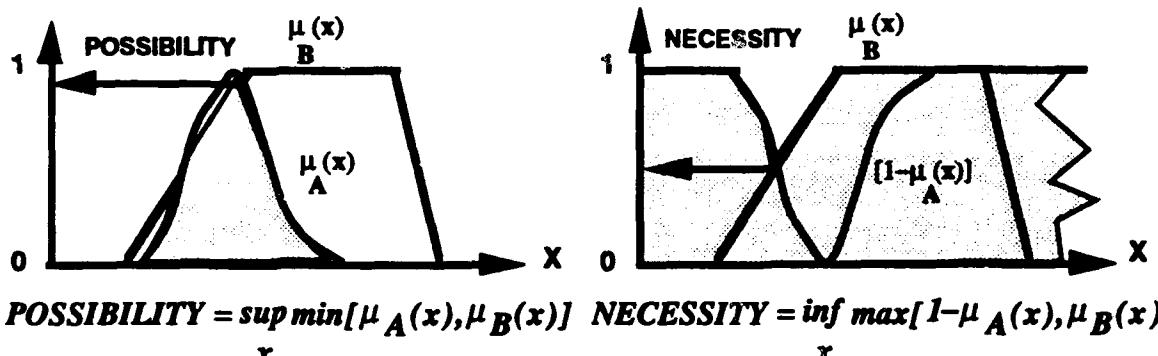
Figure 3-2. Calculation of the Necessity and Possibility of A is B

In this example, the data are used directly to calculate the bounds. In this model assessment system, the fuzzy data are first fitted by a piecewise continuous approximation and then the necessity and the possibility are constructed. This approximation increases the speed of the system by reducing the time complexity of the problem. For example, in figure 3-3a the fuzzy input is first approximated with a trapezoid, and then the possibility calculation is based on the trapezoid. The necessity is illustrated in figure 3-3b. The certainty interval that results is given by $[N,\Pi]$.
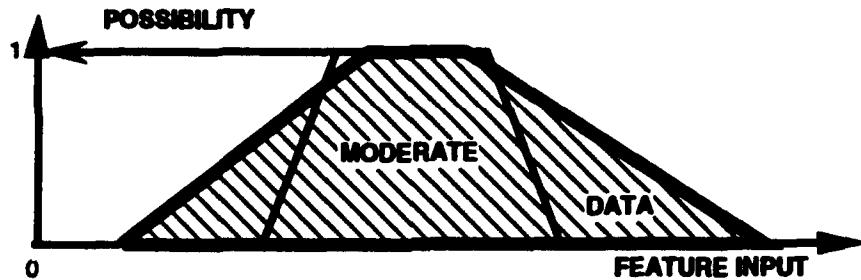
Figure 3-3a. Calculation of the Possibility from the Data for the Moderate Term
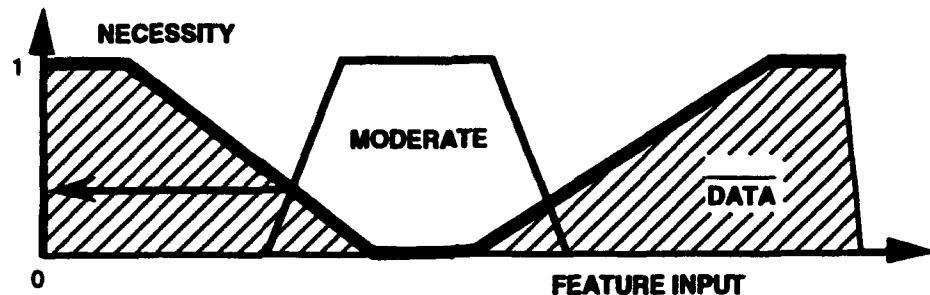
Figure 3-3b. Calculation of the Necessity from the Data for the Moderate Term

Note that when the data are singletons, then $\Pi = P = \mu_B(x)$, and the interval reduces to the single-valued representation used in the control problem example of the taxi driver. In fact, the control problem can be considered a special case of the decision problem where there is only one class; instead of estimating the certainty that each class has occurred, one uses the fuzzy rules to estimate the control parameter itself. The next section discusses the propagation of evidence through fuzzy rules when the certainty representation is interval valued.

3.1.2 Propagation of Evidence Through Fuzzy Rules

In the taxi driver control example of figure 2-3, the propagation of evidence through the fuzzy rules was described as using the minimum of the MF values for the two inputs. More specifically, the two input variables, speed and distance, were the arguments to term sets known as HIGH and NEAR. The term sets were triangular fuzzy sets and when evaluated at these inputs, the minimum membership value was used to truncate the output fuzzy set for the braking control called HARD. See the first rule in figure 2-3. In particular, the output is the rate of braking and is given by $\mu_{BRAKE}(z) = \min(\mu_{HARD}(z), \min[\mu_{HIGH}(x), \mu_{NEAR}(y)])$ so the certainty of the output can be thought of as $\min[\mu_{HIGH}(x), \mu_{NEAR}(y)]$. This concept is generalized for the interval-valued certainty measures.

In this report, the fuzzy model assessment uses interval-valued measures of certainty, and the propagation of certainty is based upon t-norms, s-norms, and the detachment operator. These operations are generalizations of the logical operations of conjunction, disjunction, and implication, respectively, and have the same *look and feel* as the operations used to propagate evidence in the control problem. The propagation of certainty and the propagation of evidence phrases are used interchangeably. In section 3.1, the certainty of the premise was derived using the necessity and the possibility. When the premise has multiple clauses, the clause certainties are aggregated to yield the premise certainty. Intuitively, aggregation means the combination of

3-3

certainties and the mathematical definitions are deferred until section 3.3. In this report, conjunctions are allowed in the premise clauses but premise disjunctions must be expanded into distinct fuzzy rules. For a more complete description of this approach, Bonissone's papers (references 9-12) are recommended to the reader. For a complete discussion of the representations of certainty, refer to reference 13.

To aggregate the premise certainty, the t-norm is used. Since the triangular norm or t-norm behaves like a logical conjunction, the strength of the premise is based on the weakest clause. The t-norm is a binary operator denoted by $T(x,y)$ where $x, y \varepsilon [0,1]$, and the minimum function is well known example. The t-norm is associative, $T(x,T(y,z)) = T(T(x,y),z)$, commutative $T(x,y) = T(y,x)$, and monotonically nondecreasing in both arguments $T(x,y) \leq T(v,w)$; if $x \leq v$ and $y \leq w$. The t-norm satisfies the boundary conditions $T(0,0) = 0$ and $T(x,1) = T(1,x) = x$, which also satisfy the definition of the logical conjunction. To apply the t-norm to the interval-valued certainty, assume that the premise is of the form $\bigcap_{i=1}^{n} \tilde{A}_i$, where $\tilde{A}_i$ is a fuzzy clause and $[a_i, A_i]$ are its interval-valued certainties. Then the premise interval certainty is given by

$$[b,B] = [T(a_1, a_2, ..., a_n), T(A_1, A_2, ..., A_n)].$$

For the special case when $T(x,y) = \min(x,y)$, the premise certainty is given by the minimum of the certainty minimums and the minimum of the certainty maximums. In general, the t-norm can be designed to reflect the data association or correlation of the premise clauses. For positively correlated clauses, the norm $T(x,y) = \min(x,y)$ is a good choice; for uncorrelated clauses the norm $T(x,y) = xy$ may be a good choice; and for negatively correlated clauses, the bold intersection $T(x,y) = \max(0, x + y - 1)$ may best capture the association (reference 10). Although the work so far has used only $T(x,y) = \min(x,y)$, the mechanism to select the appropriate t-norm for each rule is clearly in place.

The s-norm or t-conorm is a generalization of the logical disjunction or logical OR operator. The s-norm $S(x,y)$ with $x, y \varepsilon [0,1]$ has similar properties to the t-norm. These properties include associativity $S(x,S(y,z)) = S(S(x,y),z)$, commutativity $S(x,y) = S(y,x)$, and monotonicity in both arguments $S(x,y) \leq S(v,w)$ if $x \leq v$ and $y \leq w$. The boundary conditions resemble the logical OR in that $S(1,1) = 1$; $S(x,0) = S(0,x) = x$. The s-norm used in this report is the maximum function $S(x,y) = \max(x,y)$. The t-norm and s-norm are related by a generalized version of DeMorgan's law provided one uses a hard complement, i.e., $N(x) = 1 - x$. DeMorgans' laws then become $S(x,y) = N(T(N(x),N(y)))$ and $T(x,y) = N(S(N(x),N(y)))$.

The detachment operator is the mathematical mechanism needed to propagate the premise certainty through the implication operator to the conclusion. The detachment operator is defined in terms of the s-norm and t-norm operators and provides the mechanism for the propagation of evidence, not only for the interval-valued certainty representation, but also for the single-valued certainty representation. In the taxi driver control problem, the output fuzzy set of figure 2-3 can be written as $\mu_{BRAKE}(z) = \min(\mu_{HARD}(z), \min[\mu_{HIGH}(x), \mu_{NEAR}(y)])$. This output is derived from the following rule: if the speed x is HIGH and the stop light is NEAR then brake HARD. Yet there is nothing in this conclusion that includes the confidence in the rule itself. In general, if one has a rule $P \rightarrow Q$, then if $v(P)$ denotes the certainty or validity of the premise and

3-4

$v(P \rightarrow Q)$ denotes the strength or validity of the forward implication operator, then the detachment operator must relate these two quantities to the conclusion validity $v(Q)$. More precisely, the binary detachment operator $m(v(P), v(P \rightarrow Q))$ is defined so that m is as large as possible and still satisfies the constraint $m(v(P), v(P \rightarrow Q)) \leq v(Q)$ (reference 14, p. 167). That is, *the strongest conclusion you can infer without overstating the truth of the conclusion.* The validity or certainty can also be thought of as the truth, as well. If one assumes that $v(P \rightarrow Q) = 1$ then the control example is also an application of a detachment operator. The detachment operator m can be taken to be a t-norm (reference 14) and, in particular, the minimum function $\min(v(P), v(P \rightarrow Q))$ is the single-valued certainty that is often used by fuzzy expert system shells like fuzzy logic official production system (FLOPS) (reference 15). Just as the detachment operator can propagate evidence through fuzzy rules for single-valued certainty measures, the operator can also propagate evidence using interval-valued certainty measures.

The detachment operator discussed above was a function of two arguments since the validity of both the premise and the rule were real numbers in the interval [0,1]. Now the premise certainty is given by the interval $[b, B]$, and the conclusion certainty is represented by an interval as well. Moreover, it's not clear how to represent the certainty of the implication in this description. Bonissone (reference 10) uses the "necessity" and "sufficiency" of the implication operator to construct the certainty of the conclusion. The necessity is given by $n = v(Q \rightarrow P)$ and the sufficiency is given by $s = v(P \rightarrow Q)$. (Note this is not the same necessity used to measure the subsethood of one fuzzy set to another; but instead, is the classical mathematical definition of the necessary part of the if-and-only-if logical implication.) The sufficiency s is the strength of the implication in the forward direction and the necessity n is the strength of the implication in the backward direction. So the detachment operator is a vector-valued function on a vector field $m([b, B],[s, n]) = [T(s, b), S(B, 1 - n)]$ where T is a t-norm and S is its corresponding s-norm. For example, if $T = \min$, then $S = \max$ and $m([b, B],[s, n]) = [\min(s, b), \max(B, 1 - n)]$, which is the detachment operator used in the examples of section 4.

One can get a feel for this operator by simply considering the following four special cases, where the strength of the forward and backward implication operator are set at their limiting values:

1. $[s, n] = [0, 0] \Rightarrow m = [0, 1]$. Here one cannot rely on the rule in either direction so that inference mechanism has no strength. The certainty interval is appropriately given by [0,1] since the lower bound on the certainty is zero and the upper bound is 1. This simply says, nothing is known about the certainty of the conclusion.

2. $[s, n] = [1, 1] \Rightarrow m = [b, B]$. Here the implication operator is totally reliable in both the forward and the reverse direction. The strength of the conclusion is then bounded above and below by the corresponding bounds of the premise, simply because the implication operator has infinite fidelity.

3. $[s, n] = [1, 0] \Rightarrow m = [b, 1]$. Here the implication operator in the forward direction is absolutely reliable but the reverse implication operator is absolutely unreliable. The conclusion certainty interval is lower bounded by the lower bound of the premise validity, but the upper bound of the conclusion validity is 1 since no reliability can be placed on the reverse implication operator. Without the reverse implication, nothing can be inferred about the upper bound except that it is 1.

4. $[s,n] = [0,1] \Rightarrow m = [0,B]$. Here the forward implication operator is useless and so no lower bound can be established on the conclusion validity. However, the strength of the reverse implication operator allows an upper bound on the conclusion validity. One expects this in the case of *modus tollens* or backward chaining.

Note that the premise certainty interval in conjunction with the forward ply determines the strength of the lower bound, and in conjunction with the reverse ply determines the strength of the upper bound.

However, the interval interpretation is not the only one that can be placed upon the conclusion validity. The truth of the conclusion can be thought of as a linguistic variable. Figure 3-4 shows the definition of the linguistic variable TRUTH. In this figure, one sees that the semantic interpretation of the linguistic term "true" appears as the ramp function "absolutely true," as a Kronecker delta function located at a truth value of 1. Note that the term "undecided" is a constant function shaped like a uniform distribution function, much like case 1 in the above paragraph. In fact, the certainty intervals can be thought of as fuzzy sets (as illustrated in figure 1-4b and thus seen to be related to the terms in the linguistic variable TRUTH (reference 5). In fact, redefining TRUTH as in figure 3-5, the certainty intervals can be interpreted as approximations to the terms of TRUTH (reference 13). Thus the interval-valued certainties approximate the functional-valued linguistic terms of the linguistic variable TRUTH.
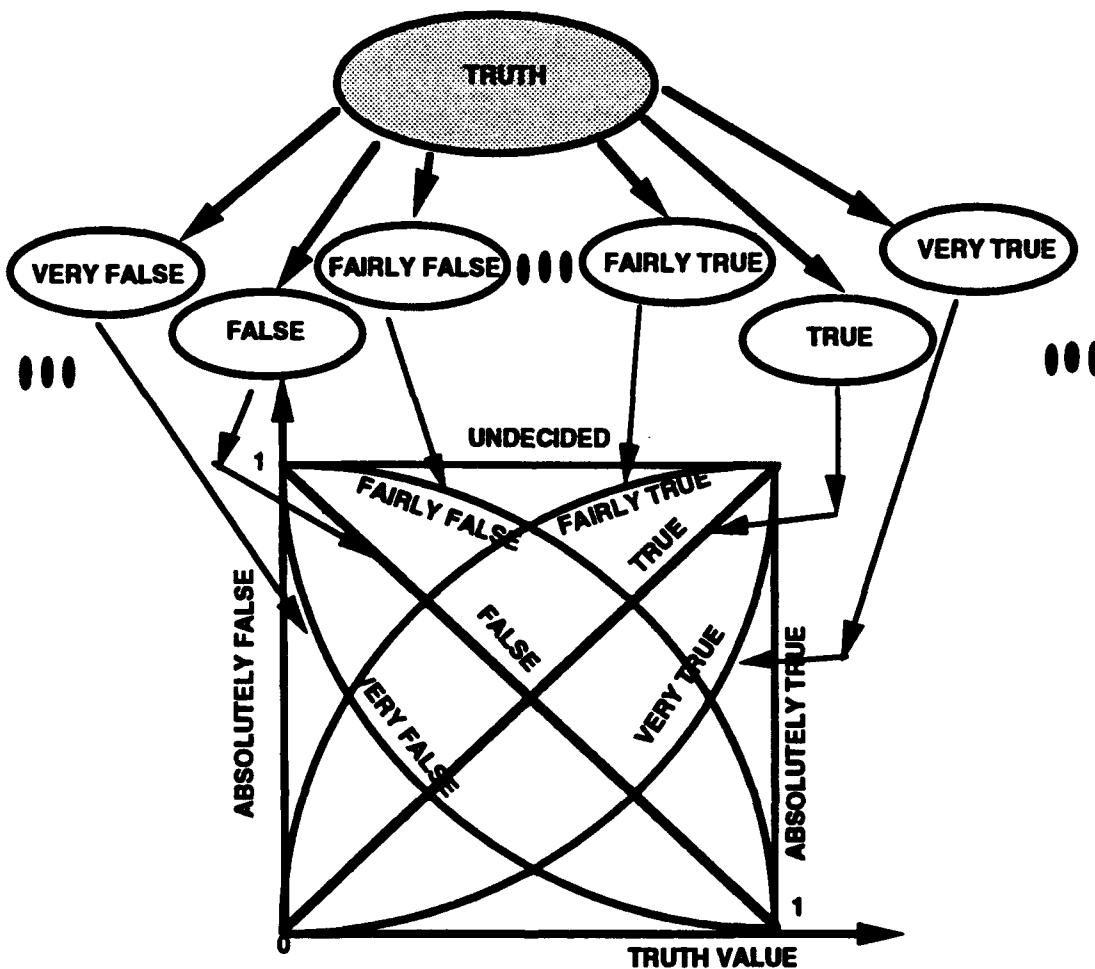

*Figure 3-4. Linguistic Variable TRUTH*
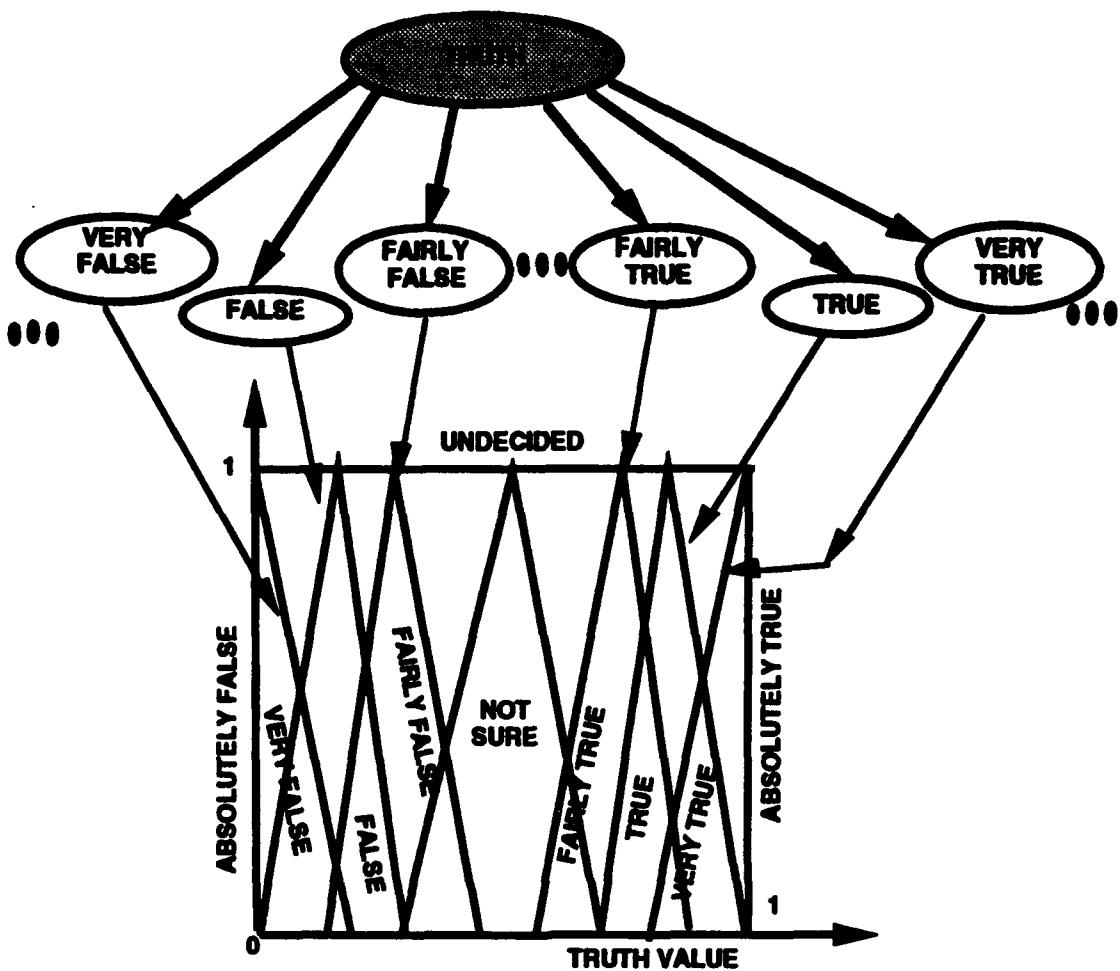
Figure 3-5. Alternate Definition of the Linguistic Variable TRUTH

3.2 DECISION SPACE

In section 2, it was pointed out that the decision function of classical pattern recognition mapped the data to singletons or single classes in the decision space. FPR generalizes classical pattern recognition by mapping to a fuzzy unit vector (fit vector), where the elements of the vectors are associated with classes and the values of the elements represent the membership in those classes. It was also pointed out that fuzzy model assessment generalizes the certainty representation so that the elements in the fit vector are now interval-valued sets and the fit vector becomes an interval-valued fuzzy set. Section 3.1 established that one could view these interval-valued fuzzy sets as second-order fuzzy sets whose members are fuzzy sets approximating the terms of the linguistic variable called TRUTH. That is, think of these intervals as approximations to the terms of the linguistic variable TRUTH. One further extension is needed. Instead of each element of the fit vector being associated with singletons in the decision space, the dimension of this vector is expanded to include one element for each member of the power set of the classes, just as in the Dempster-Shafer formulation. That is, the elements of the fit vector are members of the power set of $\{h_0, bf, pp, h_2, nl\}$, denoted $\hat{P}\{h_0, bf, pp, h_2, nl\} = \hat{P}$. In practice, fuzzy model assessment maps to a subset of $\hat{P}$. So a fuzzy rule can have a conclusion $\{bf, h_2\}$, which is interpreted as either a change in base frequency or a contact maneuver has

3-7

occurred. Associated with this decision or element of the fit vector is a fuzzy set representing the certainty interval of the conclusion.

The fact that the elements of the decision space are now members of $\hat{P}$ is a result of the fuzzy rule base. The fuzzy rules were fashioned after the compatibility maps of CMMA. The Dempster-Shafer approach used in CMMA maps the evidence or data to the frame of discernment whose initial set is the set of all possible decisions; this means that the basic probability assignment (bpa) is on $\hat{P}$. The evidence or sensor readings are mapped to the decision frame via compatibility maps. Since different sets of sensors can map to the decision frame, the evidence is combined from these different sources using the Dempster-Shafer rule of combination (ROC). Compatibility maps are very convenient in that they only ask how the current inputs can support the elements in the final decision frame of discernment. Then the support for each conclusion is aggregated using the ROC, eliminating the question of how the different sensors affect each other because this is all taken care of in the ROC. However, the ROC assumes that the sources of information or data sources are "independent," although it is not clear what independence means. Fuzzy rules also provide support for the conclusions from the data, but the support representation and the propagation mechanism are quite different.

The compatibility maps of CMMA map the bpa of the sensor frame to the bpa of the decision frame. Compatibility maps are based on compatibility relations, which are defined on the product space of the two frames. "A compatibility relation simply describes which elements from the two frames can be true simultaneously" (reference 16). A compatibility map is defined as

$$C_{A \to B}(A_k) = \left\{ b_j | (a_i, b_j) \varepsilon \Theta_{A,B}, a_i \varepsilon A_k \right\},$$

where $\Theta_A = \{a_1, a_2, \ldots, a_n\}$ and $\Theta_B = \{b_1, b_2, \ldots, b_n\}$ so support is mapped from sets in Θ_A to sets in Θ_B. The support is translated from frame to frame via the summation of the bpa that maps to a specific set, i.e.,

$$m_B(B_j) = \sum_{C_{A \to B}(A_i) = B_j} m_A(A_i),$$

where $m_B(B_j)$ is the bpa assigned to set B_j in frame B. From the formula for the bpa in frame B, it is clear that support is mapped from frame A in an additive manner. This is not the case with fuzzy rules. Fuzzy rules map support to the same sets in the decision frame, but in a different manner and in a different form. The decision space is the power set $\hat{P} = \Theta_B$, and the compatibility maps are the source of the multivaluedness of the mappings.

The fuzzy rules reflect the multivaluedness of the compatibility maps since they were fashioned from them. On a more basic level, the multivaluedness is due to the fact that single-sensor, single-measurement rules are not specific enough to reduce the ambiguity of the compatibility map or the fuzzy rule. So ambiguity is not a flaw, but a natural result of the partial observation of the data. When more specific rules are designed, the premise contains the conjunction of many measurement conditions and produces more specific decisions. Unlike binary rule-based systems, all the rules fire for each data input, but the strength of the premise satisfaction and the strength of the rule determines the strength of the conclusion. The next step is to aggregate the conclusion support from all rules, which is the counterpart to the Dempster-Shafer ROC in CMMA evidential reasoning. Aggregation is discussed in the next section.

3-8

3.3 AGGREGATION OF CERTAINTY FOR CONCLUSIONS

Conclusion certainty is combined using general aggregation operators, which behave as averaging operations (reference 4, section 2.6). Aggregation operators combine fuzzy sets in a specific manner that can be defined by a function with the appropriate properties. This function is defined as $h:[0,1]^n \rightarrow [0,1]$ and when applied to fuzzy sets defined on a common universe of discourse has the form $\mu_A(x) = h(\mu_{A_1}(x), \mu_{A_2}(x), \ldots, \mu_{A_n}(x))$. The function h is assumed to be continuous, symmetric, and monotonic nondecreasing in all its arguments. Moreover, two boundary conditions must be satisfied, namely: $h(0,0,\ldots,0) = 0$ and $h(1,1,\ldots,1) = 1$, which reflect the averaging nature of the operator. Many operators possess these properties, including the maximum and the minimum operators representing the union and intersection of the fuzzy sets in our system.

Aggregation operators are used in two different ways. The first way is to aggregate the certainty obtained from one set of data across the rules. The rules are not independent, i.e., different rules can map to the same conclusion and the data and the rules can in some sense be correlated or associated. Therefore, there may be many fuzzy certainty intervals associated with one conclusion of $\hat{P}$ originating from a single data set. These certainties need to be combined or aggregated to form one certainty interval. The second way to aggregate is across data samples. With several sets of samples, evidence will accumulate at the conclusions in the form of these interval-valued sets, and this evidence has to be combined or aggregated across the samples or across time. The second aggregation method does not have to be the same as the first. So far, this latter aggregation method has not been tested simply because the work has not progressed far enough. The first method has been tried and the next simple aggregation procedure has been tested.

For each element of the decision space, there will be a collection of intervals generated by all the rule firings. Bonissone's method of conclusion aggregation is to replace these intervals $[c_i, C_i]$ with one interval,

$$[c, C] = [S(c_1, c_2, \ldots, c_n), S(C_1, C_2, \ldots, C_n)] \quad \text{(reference 10)}.$$

When the s-norm operator is chosen as $S(x, y) = \max(x, y)$, this yields a certainty interval from the max-of-the-minimums to the max-of-the-maximums for the conclusion interval. The support of the conclusion then is the maximum support that any one rule gives to the conclusion; likewise, the possibility is the maximum of all the possibilities. This method of conclusion aggregation always yields a nonempty interval provided some rule has yielded a nonempty certainty interval. More formally, if the individual intervals are given by $\mu_i(x) = I_{[c_i, C_i]}(x)$, then the aggregation operator is given by

$$I_{[c_i, C_i]}(x) = \sup_{v \leq x} \mu_E(v) \wedge \mu_F(x), \text{ where } \mu_E(x) = \bigcap_{i=1}^{n} \mu_i(x) \text{ and } \mu_F(x) = \bigcup_{i=1}^{n} \mu_i(x).$$

The union is implemented using the maximum operator and the intersection implemented using the minimum operator. From this definition, the aggregation function h is continuous, symmetric, and a nondecreasing function of all its arguments. The boundary conditions are also satisfied. The simplicity of this method makes it practical, requiring only minimum and maximum operations. Note further, that this aggregation is conservative in that each lower bound c_i is itself the minimum of the certainties of the premise clauses and the certainty of the forward ply. So, the aggregation procedure provides a conservative necessity measure. Observe

the support that is mapped into the decision space by the fuzzy rules is not additive, as it is with the Dempster-Shafer approach.

Other alternatives to this aggregation procedure exist. Source consensus (reference 10) is one more aggregation method. Here the form of the aggregation is

$$[c, C] = [S(c_1, c_2, ..., c_n), T(C_1, C_2, ..., C_n)],$$

which, for the choice of $T(x, y) = \min(x, y)$ would yield the max-of-the-minimums and the min-of-the-maximums. This interval would be considerably shorter, and in fact could be null even when each component interval is not null. However, when all the intervals supporting a conclusion overlap, this would give a smaller interval. The necessity would be conservative, but the possibility may not. This shortened interval length tends to indicate too much knowledge of the certainty of the conclusion. Conflicting answers can yield an empty set leaving no idea of what type of support exists for the conclusions other than they may be conflicting. Bonissone points out that this is a test for conflicting evidence.

4. FUZZY CMMA SYSTEM

4.1 DATA IN THE CMMA SYSTEM

The CMMA system, as described in section 1, is a system driven by features extracted from contact tracks. These features are linear fits to data, i.e., linear regression of order one. In general, this fit could be a fuzzy regression or it could be a robust nonlinear regression, or just an ordinary least-squares regression. Robust regression procedures that can be applied to the detection and filtering of features in the model assessment problem is discussed in reference 17. The slope and intercept estimates describe a straight line and the distribution of these parameters is known, provided the errors are additive normal random deviates. Each parameter is a statistic, and its distribution is known to be normal (reference 18). The probability density function (PDF) of the derived parameters is used to represent the information as a fuzzy number, merely by renormalizing the PDF so it has a mode of one. Describing the parameters as fuzzy numbers enables the uncertainty of the linear fit to be propagated through the rule.

Figure 4-1 illustrates the overall process of fuzzifying the data from the standpoint of contact management. This solution flow diagram is also part of the tracking blackboard currently part of an independent research study (reference 19). First the track is re-initialized by detecting the start of the new segment as discussed in section 1. The new segment is modeled by a regressive fit, and the distribution of the extracted parameters is used to form the fuzzy sets labeled A and B, respectively, in the figure. Figure 4-1 shows how the jump axes are partitioned using trapezoidal term sets, which results in a waffle-iron texture in the two-dimensional feature space. In both dimensions of the jump-drift space, the trapezoids provide a fuzzy *pseudo*-partition of the space. In this figure, only the jump feature is shown along with the term sets needed to fuzzify the data.
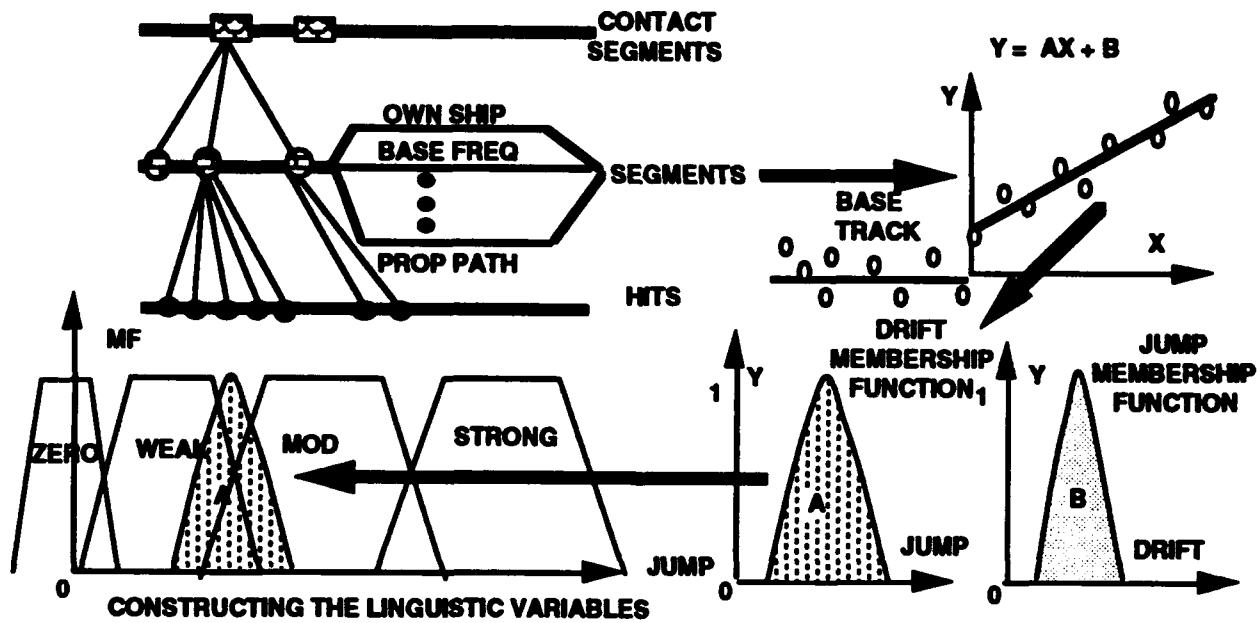
Figure 4-1. Regression Fit of Input Data, Model Fit by Fuzzy Numbers, then Fuzzify the Fuzzy Numbers from the Fit

As a practical manner, the data are not fitted directly to the term sets of the features. Instead, the data are first approximated by a piecewise continuous MF and then the possibility and the necessity is calculated from this MF. This calculation was illustrated in figure 3-3. Figure 3-1 showed how a normal data pulse is fitted by a trapezoidal approximation for one of the parameter values. This simplifies the code considerably and does so with little loss in performance and with full generality in the shape of the data. The approximation MF is represented and stored as a sequence of linear line segments. The details of this representation are contained in appendix A, so it suffices to say an MF is represented as a list of points, $((P_0,P_1),(P_1,P_2),...,(P_{n-1},P_n))$ where each point P_i is a pair of the form (x_i,y_i). The advantage of this format is simplicity, low spatial complexity, and low time complexity of the fuzzy logic operators. The formation of the possibility and the necessity was discussed conceptually in section 3. In practice, the possibility $\Pi = \sup_{x \in X} \min[\mu_A(x), \mu_B(x)]$ is calculated by first forming the intersection (minimum of the two MFs), and then finding the supremum over the resulting MF by simply looking for the maximum ordinate over the set of the segment end-points (x_i,y_i). In like manner, the necessity $N = \inf_{x \in X} \max[1 - \mu_A(x), \mu_B(x)]$ is calculated by first forming the complement of the fuzzy set A, taking its union with the fuzzy set B, and then searching for the minimum ordinate of the piecewise representation of this union. The space and time complexity of this process is clearly linear with the number of segments and thus an efficient implementation of the calculation. In figure 3-3a, calculation of the possibility was illustrated once the data had been approximated by a trapezoidal term set. The necessity is illustrated in figure 3-3b for the same trapezoidal term set.

The exploratory program for the fuzzy model assessment is written in Common LISP using the Common LISP Object System (CLOS) for the object-oriented programming aspect of the problem. Object-oriented programming attempts to encapsulate the data and the code associated with objects in the programs. Accordingly, the objects in the code correspond to natural entities in the decision process. In particular, there are objects for linguistic variables, for the conclusion, and for the data. An example is the linguistic variable object called "lingvar." Instances of this object are created for each sensor value like spherical bearing. The general form of the definition looks as follows:

```
(defclass lingvar ()
  (
  (base :initarg :base :initform 1.0 :accessor base)       ;; scaling term
  (xlow :initarg :xlow :initform -1 :accessor xlow)        ;; lower bound of range
  (xhigh :initarg :xhigh :initform 1 :accessor xhigh)      ;; upper bound of range
  (numterms :initarg :numterms :initform 7 :accessor numterms)  ;; terms number
  (terms :initarg :terms :i.iitform '(ns nm nw ze pw pm ps)
        :accessor terms)                                   ;; terms names
  (nsj :initarg :nsj :initform '() :accessor nsj)          ;; nsj names size of jump
  (nmj :initarg :nmj :initform '() :accessor nmj)          ;; nmj names size of jump
  (nwj :initarg :nwj :initform '() :accessor nwj)          ;; nwj names size of jump
  (zej :initarg :zej :initform '() :accessor zej)          ;; zej names size of jump
  (pwj :initarg :pwj :initform '() :accessor pwj)          ;; pwj names size of jump
  (pmj :initarg :pmj :initform '() :accessor pmj)          ;; pmj names size of jump
  (psj :initarg :psj :initform '() :accessor psj)          ;; psj names size of jump
  (nsd :initarg :nsd :initform '() :accessor nsd)          ;; nsd names size of drift
  (nmd :initarg :nmd :initform '() :accessor nmd)          ;; nmd names size of drift
  (nwd :initarg :nwd :initform '() :accessor nwd)          ;; nwd names size of drift
  (zed :initarg :zed :initform '() :accessor zed)          ;; zed names size of drift
  (pwd :initarg :pwd :initform '() :accessor pwd)          ;; pwd names size of drift
```

```
(pmd :initarg :pmd :initform '() :accessor pmd)          ;; pmd names size of drift
(psd :initarg :psd :initform '() :accessor psd)          ;; psd names size of drift
)
        (:documentation " Trapezoidal representation of a MF for sensor variable.")
)
```

This definition of the class called "lingvar" is used to represent terms of the sensor measurements. It represents the term sets for the features extracted from the residual data on the measurement. The linear fit to the residuals produces two components, the jump and the drift. The syntax of the term sets is $\left\{{n \atop p}\right\}\left\{{w \atop m \atop s}\right\}\left\{{j \atop d}\right\}$ if the sensor term set is not zero; for the zero term set, the syntax is $\{ze\}\left\{{j \atop d}\right\}$. The term set name grammar uses the n as negative, p as positive. The size is given as w for weak, m for moderate, and s for strong. The type of variable is j for jump and d for drift. The range of the measurement must be bounded, which incurs no loss of generality. The range is [xlow, xhigh] and the total number of terms needed to cover the measurement space is given by "numterms," currently set at seven. The variable "terms" is a list of names for these terms without the qualification of jump or drift. The important thing is not the names or the syntax, but the fact that this measurement form is captured in the object class, and each measurement has its own instantiation of this class. This class represents one of the entities in the fuzzy system diagram that describes the term sets, and is the information that is needed to fuzzify the data on input.

At present, only the positive parts of the rules are being used because the data received are folded, i.e., the absolute value of the center of the fuzzy term sets is used. This causes a reduction in the number of rules. Because the rules are essentially symmetric, then for testing purposes, the additional rules are redundant and add little knowledge to the exploratory testing. Clearly, these rules can be added easily at a later point. Of more importance, is the question of scaling. The data seen in the CMMA problem ranges over three orders of magnitude, sometimes more. This is one reason that the trapezoidal sets were used instead of triangular sets. A more appropriate scaling might be a logarithmic scaling of the data; however, the data may be either positive or negative, requiring separate scaling of the positive and negative parts and an adjustment for the values around zero. Although this decision is of no consequence for this study, it will be important when running larger data sets and handling both the negative and positive parts of the input data so the transformed data are more uniformly distributed across a finite range. When making this transformation however, the normal shape of the data will also be transformed, and how that looks in the transformed space must also be determined. The next issue addressed is when should the trapezoidal approximation to the data be made, before or after the transformation? It is probably advantageous to approximate first, then transform.

4.2 STRUCTURE OF THE FUZZY RULES

The form of the fuzzy rules in the system is similar to the simple control rules that were discussed in section 2. An example follows:

IF the drift in the spherical bearing is POSITIVE WEAK OR MODERATE
AND the jump in the spherical bearing is POSITIVE WEAK OR MODERATE,
THEN a contact maneuver has occurred.

4-3

In this format, the terms drift and jump are represented by fuzzy numbers and both the terms and their logical constructs are given by the capitalized letters. So that drift is a fuzzy number and POSITIVE WEAK is a term that is then OR'd with the term POSITIVE MODERATE. The necessity and possibility of the drift feature being a member of POSITIVE WEAK OR MODERATE are calculated and the certainty interval is formed for this clause. A similar procedure is applied to the second clause of the premise. The certainty of the premise is combined from the certainty of the clauses using the t-norm operator. The certainty of the conclusion is again an interval, which assumes that the strength of the forward and backward implication is one, that is, the *necessity = sufficiency* = 1. These variables are in the system and can be changed for all the rules or the inference engine modified to include the strengths in the rules themselves. The actual rule itself is almost self-documenting provided one recalls that in LISP the functional forms are in polish notation, i.e., (function *arg1 arg2* ...). The *cply* function is the detachment operator, and the *spbrd* function is an accessor method, which retrieves the data from the data object called sensor. The *t+* function is the fuzzy OR of the two terms POSITIVE WEAK and POSITIVE STRONG, and the *tisa* function is interpreted as *is a* or *is a member of* function. So the second and third rows of the rule produce certainty intervals, the output of the *tisa* function. The term ply is a list that contains the strength of the implication in the forward and reverse directions. For this report, ply has been set to (1 1) or (forward backward) implication strength. The conclusion, h2, is a contact maneuver. Note the *t+* and *tisa* function both utilize the piecewise continuous representation of the MFs adopted for this system.

```
(cply 'RULE_BR3 (c*                          ;; AND the predicates to form premise
(tisa (spbrd sensor) (t+ (pwd spbr) (pmd spbr)))    ;; is bearing drift positive weak or mod
(tisa (spbrj sensor)  (t+ (pwj spbr)  (pmj spbr)))   ;; is bearing jump positive weak or mod
               ) ply '(h2))                   ;; THEN conclude contact maneuver
```

The rules are kept in a separate file and used in a read-only manner. Since the form of the rules is also as a LISP function call, to fire a rule, activate it by a function call using the evaluate function.

Some of the rules presently have disjunctive forms using a function called *c+*, but this is allowed at this point *only* because the t-norm is the minimum and the s-norm is the maximum. In this one special case, disjunctions in the premises are allowed. These rules are being rewritten as separate rules. The code is designed so that any t-norm can be used in place of the min and any s-norm can be used in place of the maximum with minor modifications of the code. After all the rules have fired, the conclusions can be displayed using certainty plots. Figure 4-2 is a certainty plot for moderately high signal-to-noise (SNR) test data when a contact maneuver has occurred. Note that the single hypothesis h2 has several strong certainty intervals supporting this hypothesis. No other single hypothesis has any support. However, the multiple hypotheses have an absolute certainty consisting of the interval [1, 1]. These are to single sensor measurement rules designed to support the h2 hypothesis. Note further, all the conclusions having absolute support contain the h2 hypothesis, which confirms consistency in the rule base for this hypothesis.
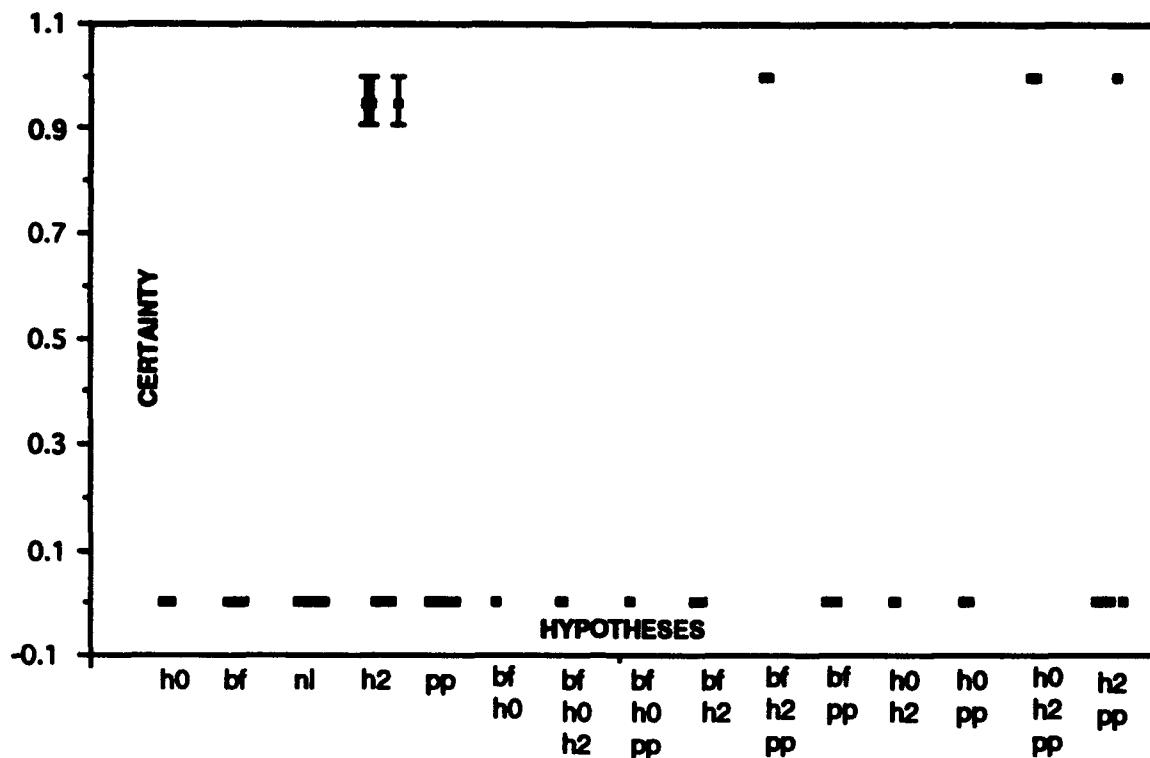
Figure 4-2. Certainty Plot of Data Designed to Test for a Contact Maneuver, Hypothesis h2

The corresponding aggregated certainty plot is given in figure 4-3. Aggregation as it is presently set up for testing purposes uses the more conservative form of conclusion aggregation, where if the certainty intervals for a single conclusion are given by $[c_i, C_i]$ then the certainty intervals for the aggregated conclusion is given by $[\max(c_1,...,c_n), \max(C_1,...,C_n)]$. Again the code can be easily generalized so the generalized s-norms can be used in place of the maximum function. This is an ideal case with strong data in all the sensor measurements. When the SNR deteriorates, then the certainty in the single hypothesis weakens causing the necessity to decrease. So the minimum support drops and the certainty intervals widen. This is illustrated in figure 4-4, by considering a low SNR version of figure 4-2. Note further that the propagation path certainty for the single hypothesis has also widened. This is because the SNR of all the sensor measurements was lowered in this example. The rule that builds the certainty for the change of propagation path alternative requires that the bearing have zero jump, and as the support of the bearing information widens (higher uncertainty in the exact bearing) the data lends support to the zero jump clause causing the pp hypothesis certainty increase. In effect, the certainty "leaks" out of the rule for which it was designed and "spills" into other rules. The weakening of the minimum support for the h2 hypothesis is caused by leakage and the increase of the support for pp is caused by the spillage. Again, this is not a flaw in the system; instead, it illustrates how the deterioration of the data uncertainty must temper the strength of the conclusions.
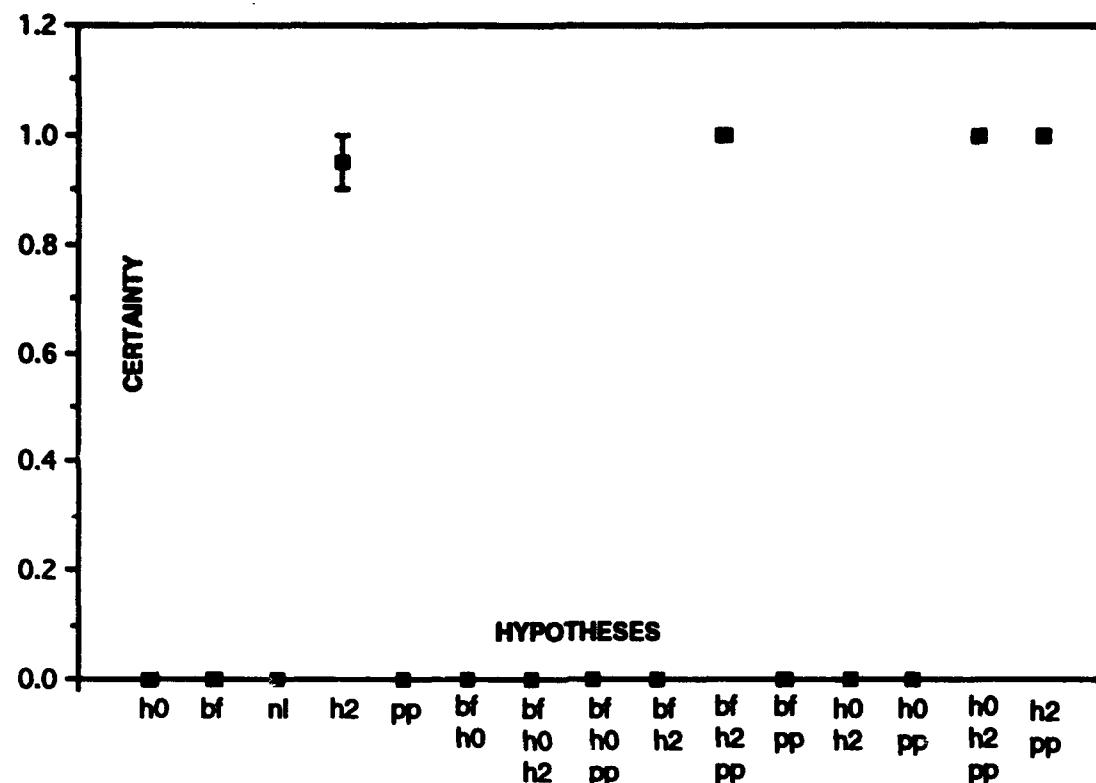
*Figure 4-3. Aggregated Certainty Plot for a Contact Maneuver, Hypothesis h2*
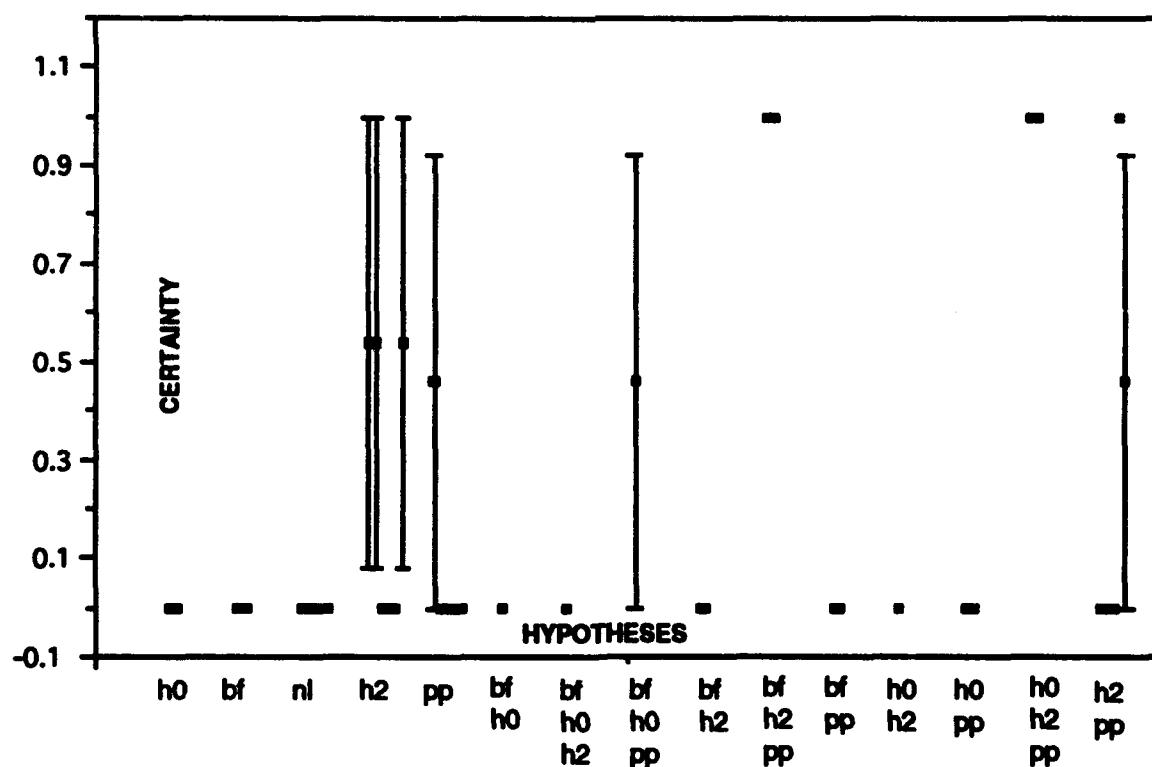
Figure 4-4. Certainty Plot of Low SNR Data Designed to Test for a Contact Maneuver, Hypothesis h2

5. CONCLUSIONS AND FUTURE DIRECTIONS

In this report, fuzzy logic has been applied to model assessment. This problem is a pattern recognition problem, which could be modeled as statistical pattern recognition or fuzzy pattern recognition. The approach investigated uses a generalized form of FPR in which the conclusions are all possible to varying degrees. Instead of modeling the degrees of possibility of the conclusions as memberships or as real numbers, the degree is modeled as an interval-valued set. The conclusions are generated by a fuzzy rule-base system using fuzzy data extracted from the residuals of linear fits to the sensor data. The fuzzy rules map the data into conclusions just as discriminant functions used in SPR except for the format of the conclusion. The fuzzy rules represent heuristics based on physical laws of motion and Doppler shifts. The system has been tested using only a few key cases to ensure that the rules and the fuzzification and aggregation algorithms work. The rule-based system works, although performance tests have not been done because no correct data set was available.

This work has generated promising results and indicated many promising directions of research. In particular, the performance of the system is dependent on the number and shape of the terms in the linguistic variables of the extracted features. The design of these terms is dependent on the clustering or learning algorithms used to create these term sets. There are numerous methods for learning the term sets. One promising technique, first suggested in references 13 and 19, applies fuzzy neural networks (FNN). These FNNs learn not only the term but also the fuzzy rules themselves. A second interesting area of study is the relationship between the fuzzy logic approach and the Dempster-Shafer approach. Clearly these two approaches are related in some sense, but the detail relationship is a difficult theoretical problem. The approximation of discriminant functions by fuzzy rules is a theoretical problem that begs for an answer. A third area of research mentioned in section 3 is the generalization of interval-valued certainty representation to term sets of the linguistic variable TRUTH. Finally, further work is needed on the aggregation of conclusion certainty, not only across rules but also across data samples. The directions for future study require both theoretical and simulation study of the issues associated with the design of this modeling problem.

6. REFERENCES

1. S. C. Nardone, D. J. Ferkinhoff, S. F. Hammel, and K. F. Gong, "Hypothesis Selection for Evidential Reasoning Systems," *1993 Basic Research Group Symposium on Command and Control Research*, National Defense University, Washington, DC, 28-29 June 1993 (UNCLASSIFIED).

2. S. C. Nardone, et al., "Contact Management Model Assessment: A System Description," NUWC-NPT Technical Document (in preparation), Naval Undersea Warfare Center Division, Newport, RI (UNCLASSIFIED).

3. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Second Edition, Academic Press, Boston, MA, 1990.

4. G. Klir and T. Foldger, *Fuzzy Sets, Uncertainty and Information*, Prentice Hall, Englewood Cliffs, NY, 1968.

5. H. Zimmermann, *Fuzzy Set Theory and its Applications*, Second Edition, Kluwer Academic Publishers, Boston, MA, 1992.

6. J. Bezdek, "Editorial Fuzzy Models - What are They, and Why?" *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1, February 1993, pp. 1-6.

7. J. Bezdek, *Pattern Recogntion with Fuzzy Object Function Algorithm*, Plenum Press, New York, 1981.

8. B. Kosko, *Neural Networks and Fuzzy Systems*, Prentice Hall, Englewood Cliffs, NY, 1992.

9. P. Bonissone and K. S. Decker, "Selecting Uncertainty Calculi and Granularity: An Experiment in Trading off Precision and Complexity," in *Uncertainty in Artificial Intelligence*, L. N. Kanal and J. F. Lemmer, eds., Elsevier Science Publishers, B. V. North Holland, 1986, pp. 217-247.

10. P. Bonissone, "Summarizing and Propagating Uncertain Information with Triangular Norms," *International Journal of Approximate Reasoning*, vol. 1, January 1987, pp. 71-101.

11. P. Bonissone, S. Gans, and S. Decker, "RUM: A Layered Architecture for Reasoning with Uncertainty," in *Proceedings 10th International Joint Conference on Artificial Intelligence(IJCAI-87)*, Milano, Italy, 1987, pp. 891-898 (UNCLASSIFIED).

12. P. Bonissone, "Using T-norm Based Uncertainty Calculi in a Naval Situation Assessment Application," in *Third Workshop on Uncertainty in Artificial Intelligence*, Seattle, WA, July 10-12, 1987, pp. 250-261.

13. P. R. Kersten, "Propagation of Evidence through Fuzzy Rules," NUWC-NPT Technical Report 10,222, Naval Undersea Warfare Center Division, Newport, RI, 1 September 1993 (UNCLASSIFIED).

14. D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, San Diego, CA, 1980.

15. J. Buckley, W. Siler, and D. Tucker, "A Fuzzy Expert System," *Fuzzy Sets and Systems*, vol. 20, 1988, pp. 129-148.

16. J. D. Lowrance, T. G. Garvey, and T. M. Strat, "A Framework for Evidential Reasoning System," *Proceedings AAAI-86, Fifth National Conference on Artificial Intelligence*, vol. 1, Philadelphia, PA, 11-15 August 1986.

17. P. R. Kersten, "Fuzzy Robust Detection and Regression in Model Assessment Problems," NUWC-NPT Technical Report (in preparation), Naval Undersea Warfare Center Division, Newport, RI (UNCLASSIFIED).

18. D. J. Ferkinhoff, J. G. Baylog, K. F. Gong, and S. C. Nardone, "Feature Detection for Model Assessment in State Estimation," NUWC Technical Report 7064, Naval Undersea Warfare Center Division, Newport, RI, 5 October 1991 (UNCLASSIFIED).

19. P. R. Kersten, "Fuzzy Expert System FY 94 IR Proposal ," Naval Undersea Warfare Center Division, Newport, RI, June 1993.

APPENDIX

CONTINUOUS LINEAR PIECEWISE
REPRESENTATION OF FUZZY SETS

One issue in implementing fuzzy logic and control is the choice of the term sets in the fuzzification and defuzzification process. Conceptually, this seems to be a trivial issue; however, for implementing logical operations and for evaluating the inclusion index, this is an important practical issue. Brute force implementations such as vector representations of the MFs have both high space and time complexity, which can slow the logic and yield erroneous results. An eloquent and efficient representation of the fuzzy sets is imperative. One such representation is discussed in this section, the linear piecewise continuous (LPC) approximation of the term sets.

In section 2 a linguistic variable was defined as a quintuple $(x, T(x), U, G, M)$ where the elements are

1. x is the name of the variable;
2. $T(x)$ is the term set of the variable x;
3. U is the universe of discourse, or domain of definition or base;
4. G is a syntactic rule for generating the name;
5. M is a fuzzy set representing the meaning of $T(x)$, M is called a semantic rule.

An example of a linguistic variable COLOR was displayed in figure 2-1. Conceptually, the fuzzy sets that define the terms in the linguistic variable are easy to represent both mathematically and symbolically. Even numerically, the MFs are quite easy to evaluate directly. In general, the result of logic operations on the MFs or semantic rules of the term sets is *not* in the same class of functions that defined the original MFs. For example, the intersection of triangle functions is again triangular; however, the union is not, provided the intersection operator is the minimum function and the union operator is the maximum function. For practicality, one wants the representation for the fuzzy sets to be closed under all the logical operations, such as OR, AND, and NOT. One set of functions that has this property is the piecewise constant functions.

For this appendix, the fuzzy set intersection and union will be implemented using the minimum and maximum, respectively. More specifically, $\mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)]$ and $\mu_{A \cup B}(x) = max[\mu_A(x), \mu_B(x)]$, respectively. The fuzzy set complement is given by $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$. The goal is closure under the three logical operations $\{\cap, \cup, \bar{}\}$, the symbol set for {AND,OR,NOT}=L, the standard set of logical operations. Another set of functions closed under L is the LPC approximation of the MFs. This representation is efficient and simple: simple since it is just a string of linear line segments connected at the end points, and efficient since the space complexity is low and the time complexity is directly related to the space complexity. The sequence of the points is represented by the list $((P_0 \ P_1)(P_1 \ P_2) \cdots (P_{n-1} \ P_n))$, where the points are themselves lists $P_i = (x_i \ y_i)$. This representation is somewhat redundant, but simplifies coding. Figure A-1 and figure A-2 contain two examples of LPC membership functions. These are trapezoidal MFs, TRAP1 and TRAP2, respectively.
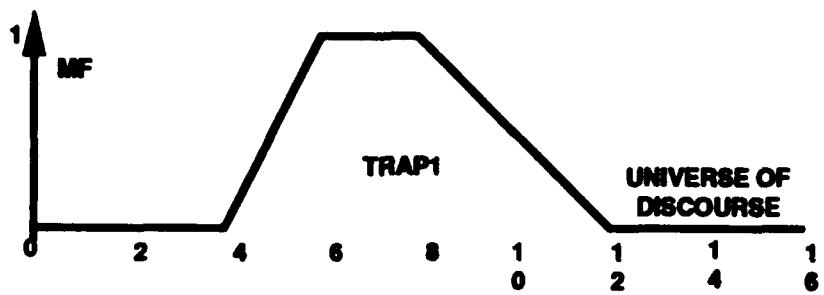
Figure A-1. Trapezoidal Membership Function, TRAP1
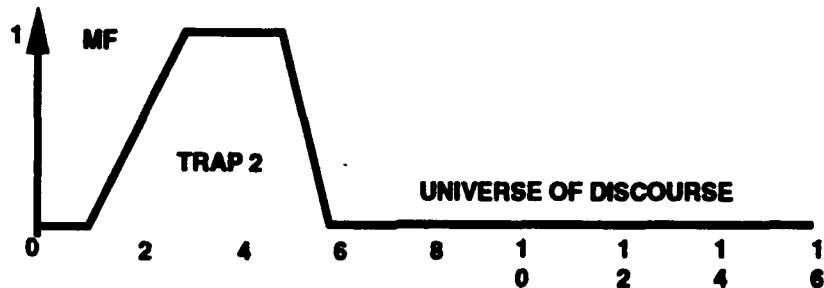
Figure A-2. Trapezoidal Membership Function, TRAP2

The sequence of x values associated with each LPC MF, induces a partition on the domain or universe of discourse. The partition can be represented by the sequence of points, $\{x_0, x_1, \cdots, x_n\}$ representing intervals of the partition given by $\{[x_0, x_1), [x_1, x_2), \ldots, [x_{n-1}, x_n]\}$. To implement binary logic operations between two MFs in an efficient manner, both MFs should have the same partition, including the same domain, which means the first and last points of the partition must be identical. Therefore, part of the implementation must include a refinement of both membership partitions so that both MFs are defined on a common partition. Moreover, this partition is still not refined enough since it needs to include all the cross-over points of the two MFs. Figure A-3 shows all the points which must be included in the partition, including the cross-over points.
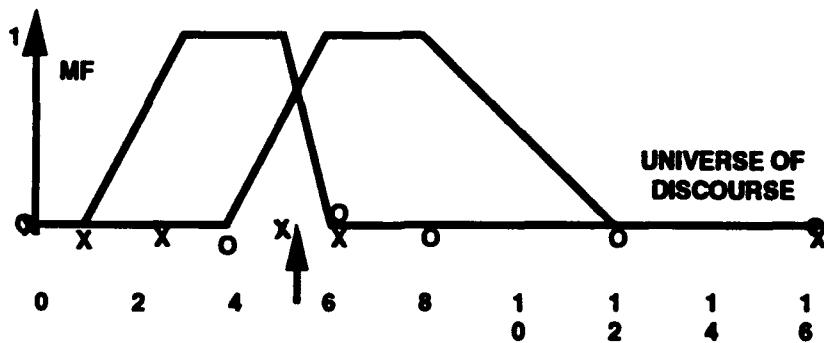

Figure A-3. Points of the Universe of Discourse Where the Partition is Made

A-2

The algorithm to carry out the binary operations is given as follows:

1. Construct the sequence of points $\{x_0, x_1, \cdots, x_n\}$ that represent the partition associated with the i-th MF, i = 1 , 2, respectively.

2. Refine the two partitions by appending the two partitions and then ordering the points to form a bag. Reduce the bag to a set by removing the duplicate points. The resulting set is a partition refinement on which both MFs are defined.

3. Construct the list of linear segments based on the refined partition for both MFs.

4. Find all the crossing points located strictly within an interval. Add these points to the refined partition and reconstruct the list of linear segments again for both functions. Both MFs associated with the logical operation now have the same partition.

5. Carry out the binary logic function on each interval of the partition constructing the linear piece on each interval and concatenating these results to form the resulting MF.

6. Simplify the resulting representation by joining adjacent intervals that represent the same linear line segment. That is, either adjacent constant line segments or continued lines with the same slope.

Figure A-4 illustrates the steps of the algorithm before collapsing the representation. The algorithm is conceptually easy to understand and moderately easy to implement in LISP.
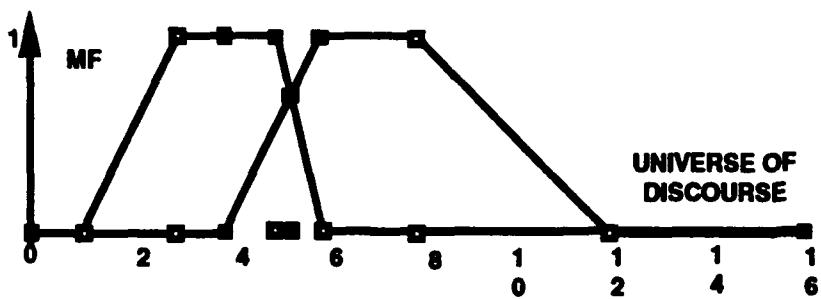
Figure A-4a. Constructing the Common Partition of the Two Fuzzy Sets
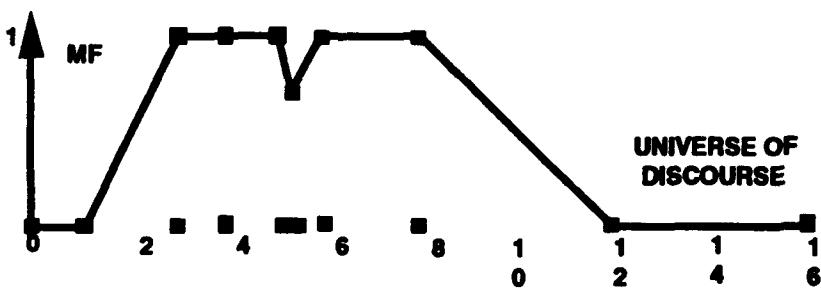
Figure A-4b. Resulting Fuzzy Set Prior to Simplification of the Partition for the Fuzzy OR Operator

The closure properties of this representation are guaranteed using the operators $\{\cap, \cup, \overline{}\}$. The minimum operation is carried out on each interval by taking the linear segment lying below the other, the maximum is implemented by taking the segment above the other, and lastly, the complement is implemented by replacing y-components with their complement $1 - y$. So once the MFs have a common partition, the logical operations are not only trivial to implement, they are also efficient.

In summary, the LPC representation of MFs is an efficient representation that includes the trapezoidal and triangular MFs. The logical operations are exact up to the round-off error effects of the computer language implementation. Moreover, the LPC representation approximates general continuous MFs and allows efficient implementation. The LPC representation retains its closure properties under unions, intersections, and complements.

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
Program Executive Officer for Undersea Warfare (ASTO-E)	1
Chief of Naval Operations (N872T, N872E2)	2
Chief of Naval Research (ONR-333)	2
Naval Air Warfare Center Weapons Division (Code 2158 — O. McNiel, J. Hodge, A. Bergman)	3
Naval Surface Warfare Center Carderock Division (Code N08 — M. Stripling)	1
Commander Submarine Force Atlantic Fleet	1
Commander Submarine Force Pacific Fleet	1
Commander Submarine Development Squadron 12	1
Advanced Research Projects Agency	2
Defense Technical Information Center	!2
Center for Naval Analyses	1